

AD-A031 975

RAND CORP SANTA MONICA CALIF

F/G 15/7

THE ARMY DEPLOYMENT SIMULATOR WITH A DATA BASE OF ARMY UNITS AN--ETC(U)

SEP 76 J H HAYES, L CUTLER

F44620-73-C-0011

UNCLASSIFIED

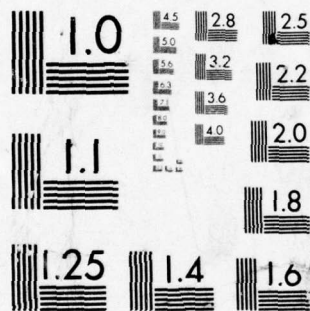
R-1893-PR

• NL

1 OF 2

AD
A031 975





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA031975

12

FG

R-1893-PR
September 1976

The Army Deployment Simulator with a Data Base of Army Units and Equipment

James H. Hayes and Leola Cutler



A report prepared for
UNITED STATES AIR FORCE PROJECT RAND

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

Rand

SANTA MONICA, CA. 90406

The research described in this report was sponsored by the United States Air Force under Contract No. F44620-73-C-0011 — Monitored by the Directorate of Planning, Programming and Analysis, Deputy Chief of Staff, Research and Development, Hq USAF.

Reports of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

The description of a flexible computer program that simulates the airlift of army units from supply bases to a threatened area. The program can be used on several types of large-scale computer. Inputs tell the computer where men and materiel are, where they must go, and what units and equipment have priority. Also specified are the number and the capabilities of each type of transport aircraft. The size and weight of each type of equipment (actual Army values) are already incorporated. The program selects efficient routes from the onload bases. Observing the assigned priorities, it allocates the aircraft to onload bases, loads them, dispatches them, and reallocates them for additional cargo. It prepares a graph of the cumulative deliveries over some specified interval. Operations of considerable complexity (an example is given in full detail) can be simulated quickly -- typically with less than 10 minutes computer time. (BK)

15 FH4620-73-C-0011

14 R-1893-PR

11 September 1976

12 170p.

6

The Army Deployment Simulator with a Data Base of Army Units and Equipment.

10 James H. Hayes and Leola Cutler

9 Interim rept.

A report prepared for

UNITED STATES AIR FORCE PROJECT RAND

A

Rand
SANTA MONICA, CA. 90406

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

1473
296 600
1B

-iii-

PREFACE

455 526

This report describes a recently completed, low-level-of-effort revision and update of an existing Rand computer program used to simulate the airlift of army units to areas of actual or potential combat. The program is one of the research tools maintained at Rand for use in Project RAND mobility studies.

A description of the original computer program was published in Rand Memorandum RM-4219-ISA (abridged), *The Army Deployment Simulator*, by William F. Sharpe, March 1965. It was developed for use in connection with policy analyses then being undertaken by the Office of the Assistant Secretary of Defense for International Security Affairs.

The original program had been prepared, of course, for the computers then in being. In the intervening years newer and more powerful computers have become available. Some changes have been necessary to adapt the original program for use in these newer machines. In addition, as with many research tools, new uses have been found for the simulation model. Over the years of use at Rand, for example, a requirement has grown for an ability to examine and to display in hard-copy form the results of studying new mobility systems and various deployment options. The present version of the Army Deployment Simulator incorporates these and other necessary changes, but leaves all of Sharpe's creative work intact.

This simulation model and its data base should be useful to planners who wish to test the capabilities of various types of currently available mobility aircraft and to examine readiness requirements for mobility forces. A user can set up the data for the deployment of army forces in a few hours, even for the most complicated cases: ten divisions, for example, with their support increments, from widely scattered locations. Less complicated and smaller deployments can be set up in correspondingly less time. Since the data herein derive from the Army's own data banks, the weights of the units listed in this report are assumed to be in close agreement with what could be expected to be realistic operational mobility requirements.

Planners should also find the model useful for testing new concepts for mobility aircraft. Once the desired design characteristics of a contemplated type of aircraft are entered into the model, the capabilities of a resultant mobility force of some given configuration and deployment can be obtained as an output for the set of conditions postulated.

The present revision and the update of the computer program were executed by Leola Cutler. The data base was completed by James H. Hayes. This work was done under the Project RAND "Strategic Mobility" research project.

SUMMARY

This report describes a computer program designed to simulate the deployment of army units via transport aircraft from tranquil locations to an area of actual or potential combat. The program is written in FORTRAN and can easily be adapted for use on any of several large-scale computers.

The program requires as inputs a description of the locations of bases (onload, enroute, and offload), the locations and compositions of various army units and prepositioned sets of equipment, the characteristics of one or more types of transport aircraft, and a statement of the priorities to be assigned to the airlift of different army units. Given the basic input data, the program:

1. Selects efficient *routes* for the various aircraft types to and from each of the onload bases.
2. Performs a detailed *loading* of each aircraft.
3. *Allocates* aircraft to onload bases.
4. Prepares a graph of the cumulative deliveries of personnel and cargo at the offload area during the deployment.

A typical deployment problem is used here to illustrate these various phases of the system.

Operations of considerable complexity can be simulated with relatively little computer time (typical cases require less than 10 minutes on an IBM 370). Executed properly, the program provides a useful tool for research on problems connected with the deployment of army units by air.

CONTENTS

PREFACE	iii
SUMMARY	v
Section	
I. INTRODUCTION	1
II. GENERAL STRUCTURE	2
III. NETWORK PHASE	5
Inputs	5
Title and Parameters (card type 00)	5
Vehicle Data (card type 01)	5
Vehicle Lists (card type 02)	9
Base Data (card type 03)	9
Link Data (card type 04)	11
Offload and Onload Bases (card type 05)	11
Aircraft Type Data (card type 06)	14
Route Selection	15
IV. LOADING PHASE	22
Inputs	22
Stock Data (card type 07)	22
First Priority Group (card type 08)	24
Aircraft Arrivals (card type 09)	24
Remaining Priority Levels (card type 10)	27
End of Case (card type 11)	27
Loading Procedure	29
Allocation of Aircraft to Bases and Priority Groups	31
V. OUTPUTS	33
Appendix	
A. INPUTS	37
B. OUTPUTS	43
C. PROGRAMMER'S GUIDE	76
D. PROGRAM LISTING	111
BIBLIOGRAPHY	163

I. INTRODUCTION

This report describes a computer program designed to simulate the deployment of army units via transport aircraft from quiet locations to an area of actual or potential combat. Although such deployments have been studied rather extensively in the past, many important aspects require further analysis. The Army Deployment Simulator saves computation time because it electronically loads, dispatches, and unloads thousands of tons of army materiel via transport aircraft. It does this quickly (typically in less than 10 minutes per run) for up to 49 onload bases and one offload base at any desired locations. Up to 10 different types of transport aircraft can be used in each simulation and about a dozen differently composed army units can be deployed. Furthermore, the program is relatively inexpensive (\$1.00 to \$10.00 per simulation, depending on the complexity), and it can give for this a variety of data outputs (detailed later) that are of use to the analyst.

To make the program inexpensive, however, a number of simplifying assumptions have been necessary; it is thus essential that the user be familiar with its logical structure; otherwise he runs the risk of using it improperly or, even worse, using it for a problem for which it is completely unsuited.

The program is designed for use on the IBM 370/158 computer. However, since it is written in the FORTRAN IV programming language (with the exception of one subroutine) it can be adapted easily for use on any one of several large-scale computers.

In this report we use an army deployment problem to illustrate various aspects of the simulator program. Section II describes the general structure of the simulator. Sections III and IV, respectively, examine the route network of the program and the aircraft loading phase. Section V discusses the program outputs. Complete listings of the inputs and outputs for the sample problem are included in the Appendixes.

II. GENERAL STRUCTURE

Figure 1 illustrates a relatively simple deployment problem using, as an example, each of the data input cards. There is a network of *bases* (Hood, Hickam, etc.) connected by links (e.g., Goose-Hickam or Hickam-Dover). The object of the deployment is to deliver certain army units to Frankfurt, the *offload base*. Highest priority is given to delivering an airborne division, second priority to delivering one infantry division and one armor division, third priority to delivering an armored cavalry regiment. To provide the materiel and personnel for these three *priority groups*, 230 transport aircraft are available to airlift them from one or more onload bases. In this case, four onload bases are indicated. A complete infantry division is stationed at Hickam, a complete armor division at Fort Hood, a complete airborne division at Fort Campbell, and a cavalry regiment at Fort Bliss. The materiel and personnel available at the four locations are summarized in seven *stock lists*, each of which indicates the quantity of various items available at the location specified. Each stock list is further identified by a *flag number* in order to facilitate processing and provide the user with greater flexibility.

Three types of aircraft are available for the deployment. Thirty C-5As arrive at Campbell, the first at hour D+0 (i.e., the first hour of deployment) and arriving every 30 minutes thereafter (0.5 hour). These aircraft are all required to move items from the stock list with flag number 1 (i.e., vehicles) on their initial delivery to the offload base. One hundred C-141As arrive at Campbell, beginning at D+10 hours, and arriving every half-hour thereafter. These latter aircraft are required to take items from the stock list with flag number 1 (vehicles) to the extent needed for priority group 1. One hundred C-141C^{*} aircraft also arrive at D+10 hours. If the 100 sorties of C-141As are more than sufficient to move these vehicles, the later arrivals (C-141As and

* The C-141C is labeled in the program "C-142" to distinguish it from the C-141A. It is introduced solely as an example of a personnel carrier.

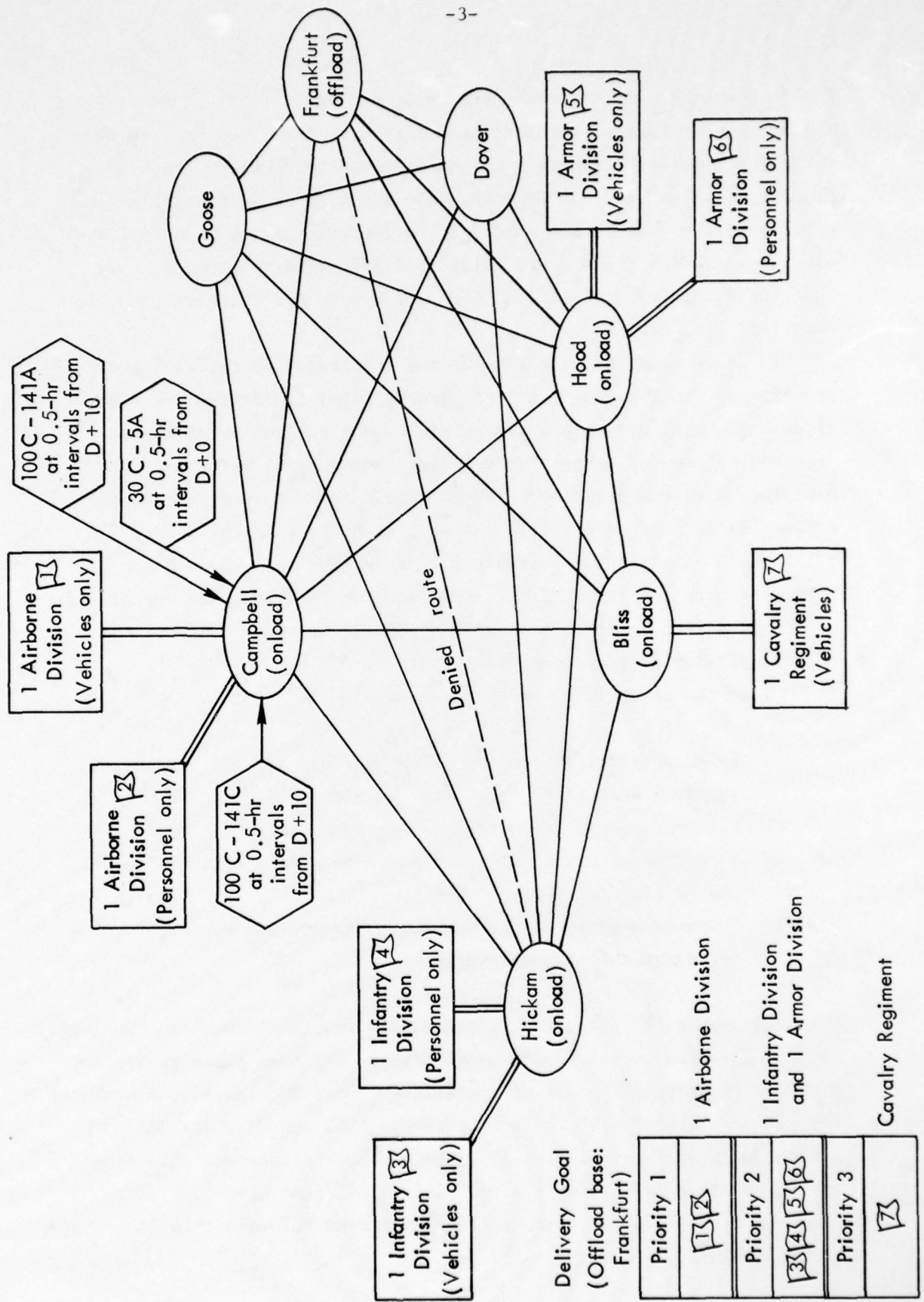


Fig. 1 — Deployment example

C-141Cs) are allowed to take items from the list with flag number 2 (passengers). Following the first aircraft arrival, the program will route and reallocate aircraft to those onload bases where the highest priority will be served. However, only stock lists with certain flag numbers may be used in the loading. For priority group 1, only lists with flag numbers 1 and 2 are eligible. For priority group 2, only those with numbers 3, 4, 5, and 6 may be used. For priority group 3, only list 7 is eligible.

To allow easy specification of the stationing of materiel and personnel, the program uses a set of *vehicle data*, indicating the dimensions and weight of various army vehicles and the number of these required by each of several units (e.g., infantry division, armored division, airborne division). In addition, one may include several *vehicle lists* that effectively isolate, with the computer's memory, all items so specified. Thereafter, the analyst may move a given army unit with the listed items included (for instance, all tanks, all helicopters, etc.), or he may exclude those items from the move for some particular analytic purpose.

Given the input data, the program:

1. Selects efficient *routes* for the various aircraft types to and from each of the potential onload bases.
2. Performs a detailed *loading* of each aircraft.
3. *Allocates* aircraft to onload bases for all sorties following the initial deliveries.
4. Prepares a graph of the cumulative deliveries of personnel and cargo for the deployment.

The program has two phases. The first performs the network and analysis, selecting routes for the deployment. The second uses these routes to simulate the actual deployment operation. For convenience, one computer run can simulate several cases of repeating the inputs required for the second phase without repeating the possibly extensive analysis required to select efficient routes. A section, below, is devoted to each phase; the problem shown in Fig. 1 is used throughout to illustrate how inputs are prepared.

III. NETWORK PHASE

INPUTS

Title and Parameters (card type 00)

Each input card must be identified by a *card type number* in columns 1 and 2. The first two cards are both type 00; the initial one provides a title for the case, the second indicates the values to be assigned to various parameters. Three of the parameters are used in the loading process; they indicate the amounts to be added to the width, the height, and the length of each vehicle in a cargo to allow for clearance. These numbers may be zero if desired.* Another parameter indicates the units of time in which the simulation results are to be recorded. An increment of four hours, for example, will result in records indicating the cumulative deliveries at the combat zone 4 hours after the beginning of the deployment, 8 hours, 12 hours, etc. Only 200 increments are available for such records, so the increment should not be made too small, since many of the deliveries would then be aggregated to the last (200th) time increment. If no increment is specified, the program will use 12-hour intervals. For large deployments, an interval of 48 hours has proved useful.

The remainder of the parameter card is normally left blank, indicating that only the standard output is desired. In certain cases, however, additional information may be necessary. By placing the numeral 1 in the appropriate position, any of the eight optional outputs described in Sec. V can be obtained.

Figures 2 and 3 illustrate the preparation of type 00 cards.

Vehicle Data (card type 01)

The program can handle as many as 200 different types of vehicle.

* Unless otherwise indicated, all numbers should be right-justified (i.e., placed as far to the right side of the field as possible) if integer values are used; fractional quantities may be punched at any position within the field by including a decimal point.

TITLE		ATLANTIC DEPLOYMENT FOR NATO		01 02		00																																																																					
05	04	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

Fig. 2-Title card

Optional Tables: Punch 1 if Table Desired																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
Width (inches)	Clearances			Hours in Each Increment	Distance Table	Vehicle Data	Priority Group Grabs	Aircraft Release Times	Priority Group	Activity at Off- Base	Loads on Initial Aircraft	Loads on All Sorties																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
	Height (inches)	Length (inches)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	30	35	40	45	50	55	60	65	67	68	69	70	71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
00	01	02	03	04					0											1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																</

Fig. 3—Parameter card

One card must be prepared for each type of equipment ("men" being one type); on the card are the name or line number of the equipment (no more than six characters), a serial number (any desired unique integer), its dimensions, its weight, and the quantities required in each of up to 11 army units. (These quantities must be integer values.) Our example uses some of the standard set of vehicle data that are incorporated in this version of the model. The units in this set (see Fig. 4 and the first page of Appendix C) are:

1. Infantry division (8-1-1)*
2. Armored division (0-6-5)*
3. Mechanized division (0-4-6)*
4. Airborne division
5. Airmobile division
6. Infantry and airborne division: initial support increment
7. Armored and mechanized division: ISI
8. Airmobile division: ISI
9. Not used (to allow independent data input)
10. Independent tank brigade
11. Armored cavalry regiment

In order to simplify the processing, personnel are treated as a vehicle subject to the following rules:

1. The first vehicle data card *must* describe personnel.
2. Clearance requirements are *not* added to personnel dimensions.
3. Personnel height is not used by the program and may be left blank.

The format for preparing vehicle data cards is illustrated in Fig. 4. Appendix A includes a complete list of the vehicle data used in the example.

*The figures in parentheses refer to the number of infantry, tank, and mechanized battalions in that order.

Vehicle Lists (card type 02)

As many as five different vehicle lists may be specified. Each one contains the serial numbers of from 1 to 100 of the vehicle types listed on the vehicle data cards. At least one list *must* be specified; experience indicates that "men" is the most frequently used one. Three such lists are illustrated in Fig. 5; one containing only personnel (list 1), one enumerating helicopters (list 2), and one including 3 types of outsize tracked vehicles on tank chassis (list 3). In some applications there will be no need to use vehicle lists; in such cases no type 02 cards need be included in the data deck.

Base Data (card type 03)

As many as 50 bases may be used in a deployment. Each base must be described on a base data card that indicates its name (up to six characters), an identifying number (any unique integer value^{*}), its ground time (described below), and its latitude and longitude. Latitudes in the southern hemisphere should be indicated by prefixing a minus sign to the *degree* figure. Longitude *west* should also be indicated by prefixing a minus sign to the degree figure. This device compensates for the inability of the computer to store "-0." Figure 6 gives the base data for the example. For -0 degrees 40 minutes, use -360 degrees 40 minutes.

The base ground times represent one of two possible sources of ground time used in calculating deployment times. The other component is based on the parameter value--ground time per flying hour--specified for each aircraft type (on card type 06). If an aircraft stops at a base it must remain on the ground for the ground time specified for that base *plus* a duration equal to the product: (ground time per flying hour) \times (flying hours since the previous stop). If, for example, an aircraft flies eight hours from base A to base B, two hours is the specified ground time at B, and the ground time per flying hour is 0.5, total ground time at B would be six hours ($2 + 0.5 \times 8$).

* The network analysis will require fewer iterations if base numbers are assigned so that the numbers generally increase as one moves from onload bases to the offload base; however, the program does not require any such numbering scheme.

The provision of these two types of ground time allows the user to: (1) employ the philosophy that all ground time, maintenance, and servicing are a function of flying time (by setting all base ground times to zero); (2) employ the philosophy that all ground time is a function of sorties (by setting ground time per flying hour to zero); or (3) adopt any desired combination of the two. Needless to say, the approach adopted may have a major impact on the results of the deployment.

Link Data (card type 04)

As indicated above, a link connects two bases. Unless otherwise directed, the program automatically computes the great circle distance between every pair of bases^{*} and places the link in the network. A link data card is entered if the user, for either of two reasons, wants the program not to compute the link distance. (1) In order to take restrictions on overflight into account, the user may enter the link distance himself. (2) He may wish to *deny* the use of a link, in which case he enters a card bearing only the air base designations, leaving the distance field blank (Fig. 7). The nonstop distance between the bases will then be the shortest possible flight pattern using links that have not been denied. If one or more of the bases indicated on a link data card was not described by a base data card, the link will be rejected, a message will be printed, and the processing will continue. If not required for the problem, link data cards may be omitted.

Offload and Onload Bases (card type 05)

To determine all routes that may be needed in a simulation, the offload base and *all* onload bases to be used in *any* of the cases to be run must be indicated on the type 05 card. Figure 8 illustrates this information prepared for our example. A blank field terminates the

* The distance is based on the arc length associated with the central angle between the two bases, assuming that the earth has a constant radius of 3433.98 nautical miles. Due to differences in the actual radius at various points, the resulting distances will only approximate the true figures. For this purpose, however, they should be more than adequate.

Base Name	Base Number	Ground Time	Latitude (+N, -S)		Longitude (+E, -W)	
			Degrees	Min	Degrees	Min
01 02	05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27			29 30 31 32 33 34 35 36 37		39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
03	BLISS	1 1	3 1	5 1	- 1 0 6	2 3
03	CAMPBL	1 2	3 6	4 0	- 8 7	2 9
03	HOD	1 3	3 1	0 4	- 9 7	5 0
03	DOVER	1 4	3 9	0 8	- 7 5	2 8
03	GOOSE	1 5	5 3	1 9	- 6 0	2 6
03	HICKAM	1 6	2 1	2 0	- 1 5 7	5 5
03	FRANKF	1 7	5 0	0 2	8	3 4

Fig. 6—Base data

Base Numbers	Distance (n mi.) Blank if Link Denied
From	To
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	
04	1 6
	1 7

Fig. 7—Link data

[illegible]

Fig. 8—Offload and onload bases

list. If more than 14 onload bases are to be used, the 15th is indicated in columns 11 through 15 of a second type 05 card, the 16th in columns 16 through 20, etc. Additional cards may be used in the same manner if required.

Aircraft Type Data (card type 06)

As many as 10 aircraft types can be used simultaneously in a deployment. Each must be described by a type 06 card. Required information includes: the aircraft name, an identifying number (any unique integer), compartment dimensions, passenger capacity, average speed, and ground time per flying hour (explained earlier). The relationship between the aircraft's payload capacity and the longest nonstop distance flown (critical leg) is described by five parameters; their meaning is illustrated in Fig. 9.

The three aircraft types used in the example are described in the standard format shown in Fig. 10. The obviously artificial compartment size shown for the C-141C is a device to insure that no cargo will be loaded in these aircraft. Under these conditions, the passenger capacity figure will be employed, if there is sufficient capacity.* This will serve to allocate C-141Cs only to passenger loads--the desired result.

The last figure on the aircraft type data card indicates the number of bases the aircraft is not allowed to use. Denying one or more bases to an aircraft type can reflect limitations imposed by runway length, maintenance facilities, etc. In some cases it may be desirable to deny the use of a base to *all* aircraft types. If, for example, the great-circle distance from base A to base C involves overflight of hostile territory, the user might enter the distance of a politically feasible route on a link data card. An alternative would be to create one or more "bases" that merely represent corners of an acceptable path. If base B were such a corner, inclusion of links A to B and B to C would

* This is one of many subterfuges that can be used to employ the program in ways not explicitly described in this document. Similar artifices enable one to simulate the deployment of tactical aircraft and ship movements.

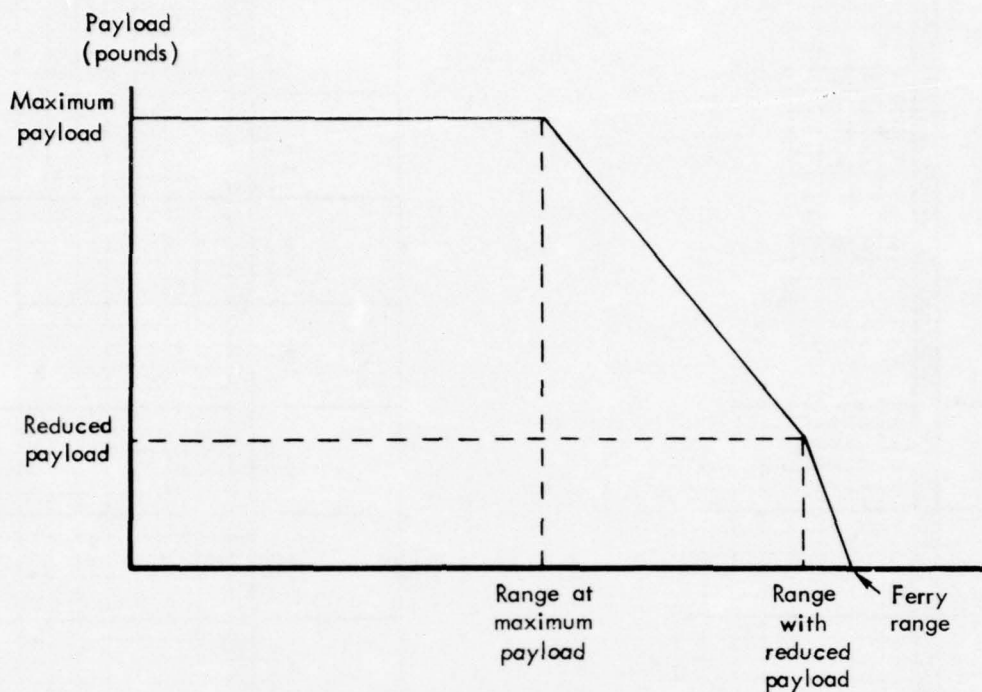


Fig. 9— Payload/range curve

give the desired result. However, since B is not really a base, it would be denied to all aircraft, to preclude their landing at it (although they may, of course, fly over it).

If one or more bases are to be denied an aircraft type, the required number of *base denial cards* must follow the aircraft type card. Since each base denial card can give the numbers of as many as 15 bases, usually only one card is required. If no bases are to be denied an aircraft type, the final field (columns 76 to 80) on its aircraft type card should be left blank and *no* base denial cards inserted after it.

The format for base denial cards is illustrated in Fig. 11, which indicates the data used for the one base where the C-141C is forbidden to land.

ROUTE SELECTION

The first step in the process of route selection involves the

[illegible]

Fig. 10—Aircraft type data

[illegible]

determination of nonstop distances between all possible pairs of bases. For links that have not been denied, such information is already calculated. For denied links, an alternative route must be calculated. The procedure used is the following. The distance between any two bases is the shortest distance along links which have not been denied. Thus the distance from Hood to Frankfurt will be the shortest of the following routes:

Hood--Frankfurt
Hood--Dover--Frankfurt
Hood--Dover--Goose--Frankfurt

When the shortest route is found, its distance is entered in a nonstop distance table. Thus for *every* pair of bases there will be a nonstop distance, involving movement along accepted links and possible flight over one or more other bases. If there is an error in link distance entered on a link data card, and some alternative route proves shorter than the distance indicated, the program will use the shorter distance. Such a choice can be detected by comparing the distance given in the (optimal) table of shortest allowable nonstop distances with that specified on the link data card.

The remaining calculations are performed separately for each aircraft type. Initially all bases denied to a given aircraft type are deleted from the table. Then a route from the offload base back to each onload base is selected. The route requiring the *minimum time* (flying plus ground time) is chosen in each case, subject to the requirement that no nonstop distance exceed the aircraft's ferry range.*

Following the selection of the minimum-time route *back* from the offload base to an onload base, the route *out* is chosen. The criterion is *maximum flow*. Figure 12 illustrates the relationship between the inputs and the route selected. For simplicity, only thirteen of all the possible routes are plotted in the fourth quadrant, which indicates

* The algorithm uses a labeling procedure of the type described in L. R. Ford, Jr., D. R. Fulkerson, *Flows in Networks*, The Rand Corporation, R-375-PR (DDC No. AD 287499), August 1962. The solution provides the minimum time from the offload base to all other bases.

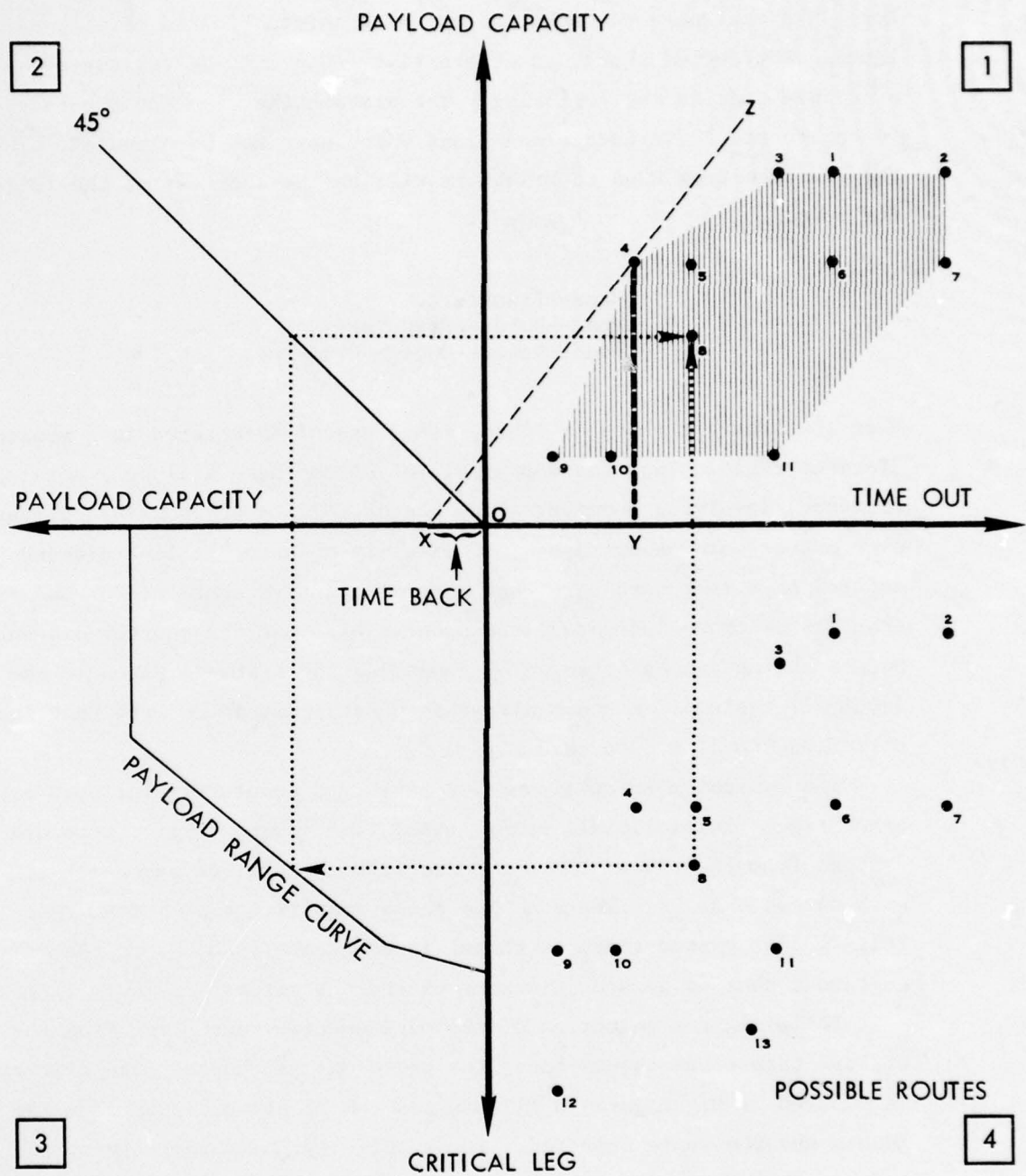


Fig. 12 — Route selection

the time required to move from the onload to the offload base (time out) versus the longest nonstop link (critical leg) on the route. By reference to the payload-range curve in the third quadrant, the payload capacity associated with each route can easily be determined, as shown for route 8.

Routes 12 and 13 are not admissible, since their critical legs exceed the ferry range of aircraft of this type. The others are admissible; however, they provide different combinations of time out and payload capacity, as shown in the first quadrant.

If just one aircraft were to be routed, only combinations of payload capacity and time, shown by the points in the first quadrant, would be relevant. But if many sorties are to be flown, by interpreting the axes as averages per sortie, many alternative combinations of routes become available--all those lying within the shaded area.* The preferred point will, however, be a single route. In Fig. 12, point X is plotted at a distance to the left of the origin equal to the time back for all routes from the onload base (using the same scale adopted for the horizontal axis to the right of the origin). Thus for route 4, the horizontal distance XY represents the round-trip time. The vertical distance from point 4 down to point Y indicates the payload carried per round trip. The ratio of these quantities we define as the *flow*: payload delivered per unit time. Since the flow of any combination will be the tangent of the angle formed by a ray from point X to the combination, it is obvious that the maximum-flow route will lie at the point of tangency between the shaded area and such a ray. As illustrated in Fig. 12, this will be a single route (in this case, route 4).

The computer searching procedure actually employed in the simulation program to obtain the maximum-flow route is quite different from that shown in Fig. 12--most important, it does not require explicit consideration of every possible route. The process is roughly as follows:

*The shaded area represents the convex hull generated by the points; it is the smallest convex figure containing all the points.

1. Exclude all nonstop distances exceeding the aircraft's ferry range.
2. Find the minimum-time route out from among the remaining combinations.
3. If no route is feasible, go to step 7.
4. If a route is found, record its flow.
5. If the critical leg on this route is less than the aircraft's range at maximum payload, go to step 7.
6. If not, exclude all nonstop distances equal to or greater than this critical leg and return to step 2.
7. (Termination procedure.) From the routes selected by the above process, select the one giving the maximum flow.

This description fails to indicate some of the aspects of the algorithm designed to conserve space in the computer's memory, but it does suggest the essence of the method.

The final step in the network analysis involves the creation of a *payload/time-out table* for each combination of aircraft type and onload base. For each such combination, a maximum-flow route out is established by the process outlined above. The bases on this route are considered the only ones where outbound sorties of this aircraft type from the onload base in question can land. However, they are not *required* to land at all of them. Often the compartment's space is filled before the aircraft's carrying capacity is reached, and an aircraft thus loaded does not *have* to stop at each of the bases specified along the maximum-flow route.* Given some payload below capacity, it would be better to avoid landing at one or more bases enroute if the time out could be reduced by doing so. To allow this, a table indicating the relationship between time out and payload actually aboard is prepared for each combination of aircraft type and onload base.

* This, of course, calls into question the efficiency of the method by which the outbound route is chosen. In some cases it may be desirable to adjust an aircraft's payload-range curve to better reflect the relationship between the weight actually loaded and the critical leg.

The algorithm is essentially the same as that used to find the maximum-flow route, with two exceptions: (1) only the bases in the maximum-flow route are used, and (2) all routes found are retained.

Two aspects of the payload/time-out table are of interest. First, if compartment size permits, an aircraft is filled to the payload capacity along the maximum-flow route; only when the actual payload falls short of this may a shorter time route be used. Second, although each entry in the table is based on a route that involves *landings* only at bases on the maximum-flow route, in some cases the implied flight path may go *over* quite different bases.

As soon as all required payload/time-out tables have been obtained, the network analysis phase is complete. In order to conserve memory space and reduce running time, the only information retained at the end of this phase is the set of payload/time-out tables and the times back to each onload base for each aircraft type.

IV. LOADING PHASE

INPUTS

Stock Data (card type 07)

After routes have been selected in the network phase of the program, the deployment cases are processed sequentially. Each case consists of a set of cards beginning with type 07 and ending with type 11.

Card type 07 is used to specify the composition of the stock lists to be made available for a deployment. There may be as many as 15 different lists. As shown in Fig. 13, each stock list is given a flag number (any desired unique integer) and must be located at one of the bases in the network. The list itself is formed from one or more components.

Each stock list is, in effect, a menu of vehicles, indicating the available quantity (which may be zero) of each vehicle type (including personnel--the first vehicle type). The flexible way a stock list is formed from one or more components is best indicated by some illustrative, hypothetical examples (not part of our sample problem).

1. The stock list will include twice the quantity of each vehicle type required by unit 1 (as shown on the type 01 cards).

Quantity	Unit	List
2	1	

2. The stock list will include 1.3 times the quantity of each vehicle type required by unit 3. If the product is not an integer it will be rounded *up* to the nearest integer.

Quantity	Unit	List
1.3	3	

3. The stock list will include 1.5 times the quantity required by unit 3 of each vehicle that appears on vehicle list 2 (card type 02). Vehicles not cited on vehicle list 2 will be excluded.

Quantity	Unit	List
1.5	3	+2

4. The stock list will include 1.5 times the quantity required by unit 3 of each vehicle type not cited on vehicle list 2.

Quantity	Unit	List
1.5	3	-2

5. The stock list will be formed from two components, with the quantities added together to form the final list. To form a stock list from two or more components it is only necessary to list the components sequentially. (Needless to say, they must all be given the same flag number and the same base number.)

Flag	Base	Quantity	Unit	List
3	17	1.5	3	+2
3	17	2	1	

The seven stock lists required for our example are described in Fig. 13. The flag numbers are used to preclude delivery of items from certain stock lists to fulfill the needs of particular priority groups. As indicated in the next section, a priority group has the same structure as a stock list. It, too, is simply a menu of quantities of different vehicle types. (Perhaps a better analogy would be a shopping list, since it represents demand rather than supply.) Even though a priority group is formed from airborne division units, for example, there is no automatic provision that keeps the program from obtaining

some vehicle (say jeeps) from a stock list that represents an armored division. Only through the use of flag numbers can possibilities such as this be avoided.

In this example, different flag numbers have been assigned each stock list. This is not necessary. One might, for example, assign flag number 1 to all stock lists representing airborne units, and flag number 2 to all lists representing infantry units, and flag number 3 to those representing stocks of prepositioned vehicles. Priority groups representing airborne units could be filled from lists with flag numbers 1 and 3; those representing infantry units from lists with flag numbers 2 and 3.

First Priority Group (card type 08)

The first priority group is composed of those flags that are needed first at the offload base. It states the quantities of the various vehicles/men most required--a set of requirements to be given first priority in the deployment. These needs can only be filled, however, from stock lists that have one of the flag numbers indicated for the priority group. (Other priority groups are listed on card type 10.) The first priority group data for the example are shown in Fig. 14. Although only one component appears in the example, any desired number can be specified. Flag numbers must be repeated on all cards.

Aircraft Arrivals (card type 09)

Any aircraft to be used must be entered into the network at some specified onload base. The aircraft arrivals are specified with type 09 cards--each of which describes the arrival of a group of aircraft of one type at its given base. The program cannot accommodate more than 1,000 aircraft in all; an aircraft group may include from 1 to 1,000 aircraft. The aircraft within each group arrive at a constant rate following an initial delay. If no delay is desired, the "hours before first arrival" may be specified as zero. If all aircraft within the group arrive concurrently, the "time between arrivals" should be set at zero.

Flag Number	Base Number	Quantity	Unit	List	
				+	No.
0102	04070809	10111213141516171819202122232425		29303132333435363738394041424344454647484950515253545556575859606162636465666768697071727374757677787980	
07		1	1	-	1
07		2	1 0	+	1
07		3	1	-	1
07		4	1 0	+	1
07		5	1 3	-	1
07		6	1 0	+	1
07		7	1 1	-	1

Fig. 13—Stock data

Flag Number	Flag Numbers Allowed		Quantity	Unit	List	
	1	2			+	No.
0102	04070809	1011121314151617181920212223242526272829303132333435			394041424344454647484950515253545556575859606162636465666768697071727374757677787980	
08		1	1	4	-	1
08		2	1 0	4	+	1

Fig. 14—First priority group data

In describing a group of aircraft arrivals, it is essential to indicate two flag numbers of stock lists at the onload base in question.* As long as items needed for the first priority group remain in the stock list with the first flag number specified, they will be selected and will be loaded on any aircraft type that is available and that can carry the equipment. If none remain, however, the stock list with the second flag number specified will be selected. Two flag numbers must be indicated for each group of aircraft arrivals. If only one stock list is to be used, its flag number can be entered in both fields. This has been done for the C-5A arrivals in the example, as shown in Fig. 15.

There is no restriction on the manner of grouping aircraft arrivals. Aircraft of a single type can arrive at several different onload bases (there would be a separate arrival card for each). Moreover, several arrival groups may use the same aircraft type and onload base. This allows the user to represent virtually any desired arrival pattern--rectangular or nonrectangular. It can also be used to avoid certain difficulties that may arise when very large groups of aircraft arrivals are specified.

The program loads and dispatches aircraft for their initial sorties in the order listed on the aircraft arrival cards. Although *within* a group of arriving aircraft the loading proceeds according to arrival time, *between* groups it typically does not. Moreover, all aircraft are loaded for their initial sorties before any recycling from the offload base is begun. For these reasons, the manner of grouping arrivals may affect the results of the simulation.

The aircraft are loaded with items of priority group 1. This is done to the extent that items are available in the stock lists allowed the aircraft group. When loaded, an aircraft is sent to the offload base. If no required items are available on an initial loading, the

* If priority groups are used, it is of the nature of the model that first-priority materiel be loaded on all types of aircraft. In large deployments this may lead to large numbers of aircraft arriving at one base. This is an artificiality but does not affect the total deployment time.

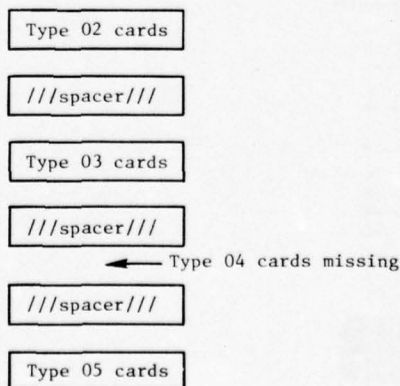
aircraft is sent to the offload base empty and a message printed. When all arrivals have been processed, three (internal) records of the results of the first set of sorties are available. The first gives a list of the aircraft, indicating the time at which each completed the requisite ground time at the offload base, and thus became available for redeployment. The second record indicates the number of personnel delivered to the offload base in each of the 200 time increments. The third gives the weight of all other types of vehicles delivered in each of the time increments.

Remaining Priority Levels (card type 10)

If more than one priority level is desired, the necessary number of type 10 cards must be included (Fig. 16). The components of subsequent priority groups are specified much as those are of the first priority group. Priority groups are listed in order, with the higher priority groups preceding the lower. A change in priority group is indicated by inserting a blank card between the component cards. If only one priority distinction is desired, that priority group is described on the type 08 card and type 10 cards are omitted.

End of Case (card type 11)

In setting up the cards required for one or more deployment cases it is essential that *spacer cards* be included. One of these cards, which are completely blank, must be inserted after each group of cards of a given type. If type 04 cards are omitted, two spacers are used as shown below:



Base Number	Flag No. of		Aircraft type (number)	Quantity in group	Hrs. before First arrival		Hours between arrivals	
	1st Stock List	2nd Stock List			First arrival			
06 07 08 09 10	11 12 13 14 15 16 17 18 19 20	1	1	3	0			
09	1 2	1	5	0				
09	1 2	1	1 4 1	1	0			
09	1 2	2	1 4 2	1	0			

Fig. 15—Aircraft arrivals

[illegible]

Fig. 16—Remaining priority groups

After the last type 10 card (or, if there are none, after the last type 09 card), there must be a spacer followed by a card with 11 punched in the first two columns to signal the end of a case. If additional cases are to be run, they should follow the type 11 card; the first card following it will be a spacer, followed by the type 07 card describing the first stock list for the new case.

A complete listing of the input data for the deployment example is included in Appendix A.

LOADING PROCEDURE

The procedure followed in loading aircraft during the deployment simulations gives priority to vehicles over passengers and, among feasible vehicles, selects the widest first. Throughout, loading assumes rectangular parallelepipeds--both for vehicles and for aircraft compartments. However, a user requiring a closer approximation to realism could alter the standard method of computing the dimensions of aircraft compartments.

Whenever an aircraft is to be loaded, two lists of vehicles must be considered: the list of vehicles still required for the current priority group and the list of vehicles still available in the stock list to which the aircraft has been assigned. At each step in the loading process the only vehicles to be considered are those appearing on *both* lists. As soon as a vehicle is selected for loading, both the priority group list and the stock list are adjusted accordingly.

The first vehicle loaded into an aircraft is the widest feasible one, that is, it is neither too wide, too long, too high, nor too heavy for the compartment. When selected, the vehicle is placed in the forward left-hand corner of the compartment, as shown in Fig. 17. Next, the unoccupied space to the right of the vehicle (in this case, the area BGJE) is considered. Again, the widest feasible vehicle (taking into account the presence of vehicle 1) is selected and placed in the forward left-hand corner of the available space. The program then moves to the right once more, attempting to load area CHJE. Eventually, no vehicles can be loaded in the space under consideration. At this point the program moves to the rear of the farthest forward vehicle

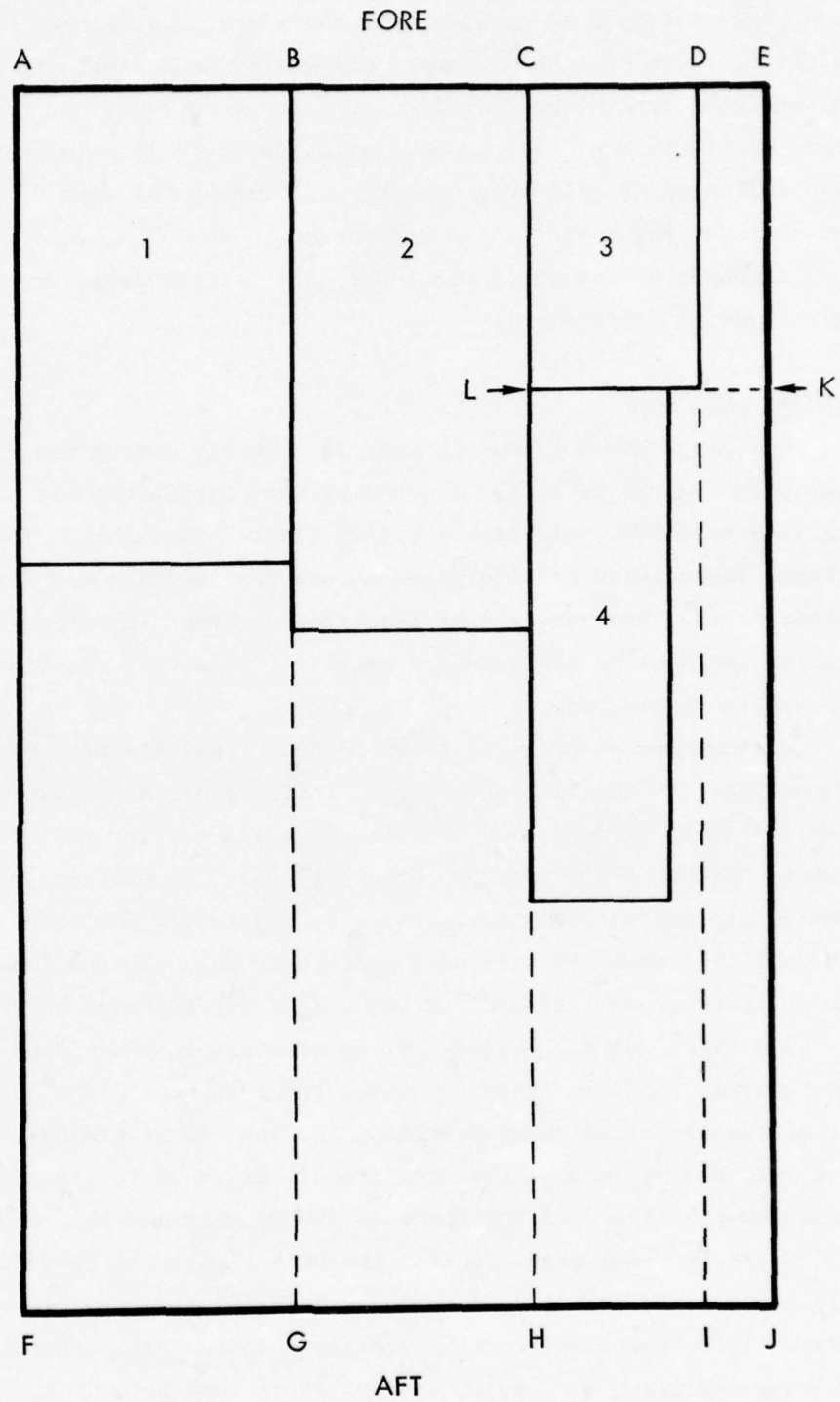


Fig. 17—Vehicle loading

but aft of vehicle 3 and considers the space behind it (LHJK). Loading proceeds in this manner, from front to rear and, along each successive cross section, from left to right, until no further vehicles can be placed in the compartment.

As vehicles are loaded, there will often be space left along one or both side walls of the aircraft compartment. Whenever such space exceeds the width of a passenger (as specified on the first type 01 card), its length along the wall is recorded. After all vehicles have been loaded, the total length of such space is divided by the length of a passenger to estimate the space capacity for passengers. This figure is compared with that obtained by dividing the unused payload capacity by the weight of a passenger. The smaller of the two determines the number of passengers carried.

An exception to this passenger-loading method occurs whenever it is impossible to load any vehicles in an aircraft, either because no vehicles remain in the priority group or the stock list, or because none will fit the compartment. The specification of a 1 cubic inch compartment for an aircraft is one way of insuring that it will be treated as a passenger aircraft. Whenever it is impossible to load any vehicles in an aircraft, the passenger capacity specified on the type 06 card is used, unless payload limitations require a reduction.

If, at loadings other than the initial one, it is impossible to load an aircraft at all (i.e., with either passengers or vehicles), it is not assigned in the manner considered, being allocated instead either to some other stock list or to another priority group.

ALLOCATION OF AIRCRAFT TO BASES AND PRIORITY GROUPS

After all of the aircraft arriving initially have been processed, a record of the time when each aircraft becomes available for redeployment from the offload base remains. The aircraft are rerouted to various onload bases in accordance with the following rules:

1. If *any* eligible onload base has *any* item needed for a priority group, and the aircraft in question can carry it, that aircraft will carry the priority materiel only; it cannot accept loads of two priority levels.

2. If more than one eligible stock list contains items needed for a priority group, and an aircraft can carry them, the list located at the onload base with the greatest flow capacity will be chosen. (The flow capacity of a base is simply the figure calculated for the maximum-flow route from the base to the offload base.)

The first rule indicates the operational meaning of the term priority group; the second suggests that the achievement of a maximum-flow pattern of deliveries within each priority group is pursued in allocating aircraft as well as routing them. For any given priority group, sorties are processed chronologically as aircraft become available at the offload base. All processing (loading and delivering) is done one priority group at a time. When an aircraft is found to be unable to deliver any items for the priority group being processed, it is not redeployed; instead it remains at the offload base, but the time when it was first available is recorded. As soon as processing of the next priority group begins, all passed-over aircraft are re-investigated and deployed as of the time recorded (if they can carry some of the new priority group items).

V. OUTPUTS

The major output of the program is a graph summarizing the results of the deployment simulation. Figure 18 shows such a graph, which is normally read by turning the page 90° counterclockwise. For each time increment from the one preceding the first delivery group through the one in which the last delivery takes place, the graph indicates the cumulative percentage of the requirements delivered; personnel is shown by dots and cargo by asterisks. Each hyphen at the bottom of the graph represents roughly two percent. For greater precision, the data are also printed beside the graph.

Needless to say, the aggregation of as many as 200 different vehicle types into a simple measure based on weight obscures many important details of the deployment. Some meaning, however, may be attached to the graph. If one subscribes to the idea that personnel and equipment must be available in exactly the prescribed ratio to be useful, the better measure of effectiveness would be the lower of the two curves in each time increment.

In addition to the summary graph, certain other information is provided as standard output: routes selected, the composition of the stock lists and priority groups, the number of sorties by aircraft type, etc. In addition to this standard output, eight optional types of information can be obtained by appropriate entries on the parameter card (type 00).

1. *Vehicle data.* This option provides a table containing all the information from the type 01 cards; the dimensions listed *include* the clearance requirements.
2. *Distance table.* This option provides a table giving the shortest allowable nonstop distance for each possible pair of bases. For links not specified on a link data card, the great circle distance will be shown. For others the distance will be the shortest over admissible links.

CUMULATIVE DELIVERIES OF MATERIEL AND PERSONNEL FROM ALL PRIORITY GROUPS TO OFFLOAD BASE

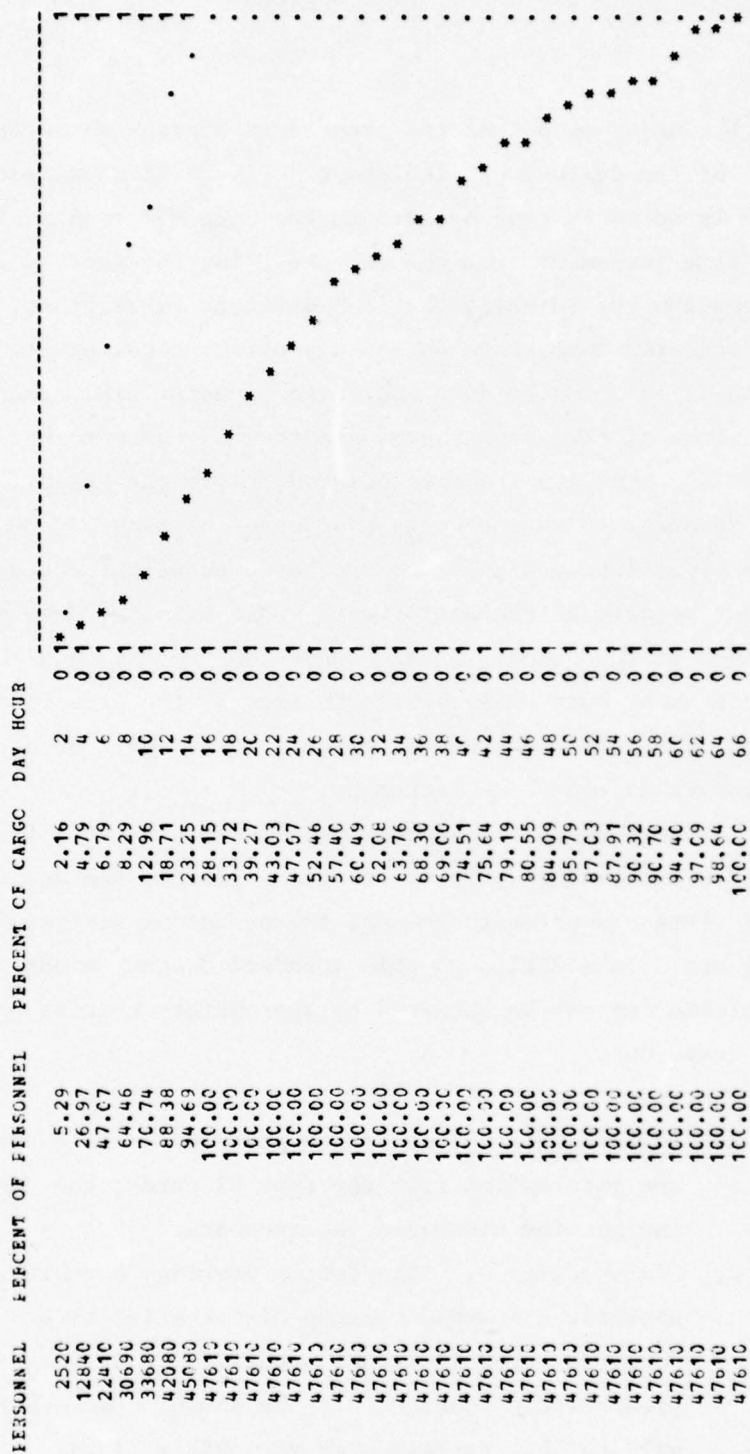


Fig.18—Output summary graph

3. *Priority group composition.* This option provides a listing of each vehicle present in each of the priority groups and the quantity required.
4. *Priority group graphs.* This option provides a cumulative arrival pattern graph for each priority group. The form is similar to that used for the cumulative delivery pattern.
5. *Activity at offload base.* This option provides a record of the arrivals and departures of aircraft at the offload base during each time increment, and the number on the ground at the base at the end of each time period.
6. *Aircraft release times.* This option provides a record of the time when each aircraft in the deployment completed its last sortie (and thus became available at the offload base for use in another operation).
7. *Loads for initial sorties.* This option provides a record of the loads of the initial sorties of each of the aircraft used in the deployment.
8. *Load for each sortie.* This option provides a record of the load carried in each of the sorties flown in a deployment. It permits calculation of average aircraft loads. However, in large deployments where the sorties may run into the thousands, some care must be exercised in using this option.

The full output for the example is shown in Appendix B, which illustrates the form in which the information is presented.

-37-

Appendix A

INPUTS

The equipment lists (the 01 cards) are for ten different kinds of army units:

Columns 37-40:	Infantry division
Columns 41-44:	Armored division
Columns 45-48:	Mechanized division
Columns 49-52:	Airborne division
Columns 53-56:	Airmobile division
Columns 57-60:	Infantry and airborne ISI
Columns 61-64:	Armored and mechanized ISI
Columns 65-68:	Airmobile ISI
Columns 69-72:	Unused
Columns 73-76:	Independent armored brigade
Columns 77-80:	Armored cavalry regiment

The user must bear in mind that the personnel figures given in the 01 card list for this example are 10 percent of the total personnel strength. Therefore, the multiplier 10 must be used as described in Sec. IV in the instructions on the 07 card. Other data for other runs may not require this device.

Another note is necessary. The model permits only 200 different types of equipment to be used. Since between 600 and 700 different equipment line items appear in the units listed above, equipment items of like dimensions and weight are aggregated under one item. As a result, if a reader checks a particular item, he may find that the number shown on the 01 card exceeds allowances for that unit or, in some cases, the line is an invention to encompass a group of items of similar weight and dimensions (YA0001).

The artifice described above does not affect the loading units or the total weights, which are very close to actual operational requirements.

ATLANTIC DEPLOYMENT FOR NATO														
00	00	4	0	0	48	1	1	1	1	1	1	1	1	1
01	MEN	1	30		20	24016691679164914131773182717711605								421 325
01	K31795	4	113	156	697	6161	37	14	14	88	287	96	28	23
01	C20414	5	73	66	264	30946	8	18	14					4 5
01	P39241	6	117	135	268	36345	4	4	4			12	18	17
01	P39378	7	129	149	577	60800	3	3	3	3	1	39	54	40
01	L43664	8	144	112	340	87700	6	16	12			3		3
01	W76816	9	136	131	233	48960	9	8	7			80	110	78
01	X48914	10	102	123	269	22960	6	6	6	1		41	82	36
01	K90188	11	99	129	269	16860	12	62	24					
01	S70243	12	97	121	593	17435	1	1	1			3	5	
01	S70517	13	115	67	416	16285	12	10	10			88	47	46
01	S72024	14	108	98	352	13980	38	37	37	3	1	86	48	98
01	S72983	15	97	109	377	15470	18	30	25			52	161	108
01	S73394	16	128	94	479	45870	2	6	6			4	4	2
01	S74079	17	99	146	347	15850	9	9	9	7		2	2	2
01	S74400	18	97	134	320	16343	9	10	9	8	4	11	6	5
01	X40009	19	96	81	265	15180	453	473	437	122	90	697	691	624
01	X40146	20	96	81	279	15570	392	247	238	70	29	271	259	68
01	X40283	21	96	81	329	14876	4	4	4	1	1	6	6	6
01	X40831	22	98	86	304	20910	42	179	159			32	35	26
01	X40968	23	115	87	320	21771	93	161	155			4	8	6
01	X43845	24	98	91	289	23847	45	22	22			57	24	52
01	X59326	25	98	96	264	19387	80	93	85	18	1	308	310	367
01	X59463	26	102	86	280	20107	1	1	1			1	1	5
01	X59600	27	115	111	288	30612	2	6	6			2	6	2
01	X59874	28	115	111	290	30082	12	10	10			37	51	35
01	X60696	29	117	99	360	33874	1	1	1			6	7	9
01	X62340	30	100	130	265	15760	79	72	74	2		35	31	38
01	X62477	31	100	130	279	16102	8	8	8			7	7	4
01	X63299	32	99	106	355	34820	46	41	48			38	44	36
01	K27486	33	98	98	391	26560	8	8	8					
01	L45534	34	120	148	513	42680	4	4	4					
01	A93125	35	111	98	249	29960	27	27	27			30	6	27
01	E56577	36	147	130	354	110560	3	8	8					
01	J96694	37	106	105	192	24200	24	24	24					
01	K56981	38	124	108	265	57630	4	12	12			12	12	
01	K57666	39	125	110	353	49048	54	54	54			18	18	18
01	R50544	40	124	115	254	48448	18	13	61			6	4	9
01	R50681	41	135	118	321	107600	10	37	27			6	5	4
01	V13101	42	144	129	325	97000	54	324	216			17	17	117
01	T10275	43	97	129	322	20000	1	1	3			4	9	8
01	T13152	44	96	113	335	32780	1	1	1	1		7	5	5
01	P43077	46	102	89	203	15470				3	1	3		9
01	J74920	47	91	90	309	20940				4		9	9	9
01	X63162	48	96	101	303	23960				11	3	1	1	1
01	S12575	50	120	54	162	6540						2	24	24
01	W76268	51	86	84	155	20290						4	96	96
01	J35698	53	97	118	313	26430			2			4	8	7
01	K30378	54	142	141	611	17660						96	129	66
01	D12087	55	100	84	192	20125	R9	358	397			43	34	79

01	D11401	56	75	91	168	13503	74	175	179			67	7	50		42	51
01	D11538	57	101	104	192	22415	22	138	145			15		25		29	10
01	E02670	58	97	51	188	2720	4	6	4	8		24	20	29			
01	A94030	59	95	56	148	2750	102	105	105	6		17	29	7		42	
01	A95263	61	85	51	150	1640	2	2	2			18	18	3			
01	A95537	63	74	50	147	1350	549	523	330	505	351	210	261	208		80	
01	A95811	64	83	55	166	2670	638	659	632	79	76	647	567	519		140	
01	A98425	65	82	81	162	2530	120	116	119	79	26	164	198	103		25	
01	A38639	66	66	53	142	2895	58	37	34	1		4	4	4		11	3
01	A38776	67	83	91	200	6900	42	41	41	484	255	46	38	43		21	1
01	A39735	68	76	65	186	5660	687	469	406	484	259	273	624	332		85	62
01	A39872	69	76	65	191	5836	93	151	125	192	147	55	50	40		31	
01	A57271	70	96	92	264	14600	21	20	18	2		33	37	30		5	
01	A57408	71	96	92	278	15022	2	2	2			14	15	15			21
01	A59052	72	94	82	229	11703	2	2	2			11	17	19			
01	A59189	73	94	82	243	12103	1	1	1							1	
01	A60833	74	64	53	132	24500	1061	905	914	818	670	573	772	237		222	112
01	A57524	76	57	70	284	4760	54										
01	V12826	82	126	105	277	46580										51	
01	V12863	83	143	117	252	95500										51	
01	B12482	85	42	42	225	4540						1	1	1			
01	B18373	87	97	66	196	7510						7	16	10			
01	B63711	90	118	44	154	9110						1	1	3			
01	C30497	108	46	54	73	1675						58	155	11			
01	K29660	111	128	150	626	7475	15	9	9								
01	D11049	113	100	77	232	16774								42	48		
01	D02533	117	98	47	188	3675						58	18	18			
01	A40960	118	84	90	166	4905						17	33	49		3	
01	A59626	119	84	84	145	16280						2	2	2			
01	A59631	120	81	86	122	9385						4	4	4			
01	A73352	122	78	67	163	10660						2	1	6			
01	A84031	123	68	64	105	3260						26	37	14			
01	A06424	124	120	111	434	5800						4	4	4			
01	A40474	125	130	153	212	84330								4			
01	A43364	126	124	122	182	42630						4	4	4			
01	A43414	127	44	34	196	3356						14	16	14			
01	A43429	128	100	118	459	65880								1			
01	A50721	129	108	144	410	72030						1	1	1			
01	A51132	131	118	143	495	61450						1	1	1			
01	A27844	134	100	122	340	24760						2	2	3			
01	A28412	135	99	110	290	21300						6	6	7			
01	A29945	136	99	126	355	37100						2	2	3			
01	A40261	138	98	115	198	12620						1	1	2			
01	A55186	139	95	115	199	7070						3	1	3			
01	H01855	140	99	131	394	18080						9	15	13		2	
01	H01857	141	99	131	327	21390						6	10	10			
01	J30766	144	102	134	417	29630						1		3			
01	J31451	145	100	139	261	57304						2		6			
01	J42100	151	83	68	176	4680						54	81	27	47		
01	J74652	153	95	124	313	28250						13	17	13			
01	J97230	155	124	108	443	59200						12	12	12			
01	K04697	156	25	24	137	7000						4		5			
01	K25215	160	88	104	169	8530								2			
01	K57667	163	125	110	355	49048						6		12			
01	L21437	164	68	56	144	1830						4	5	1			
01	L45808	165	96	92	117	4265						24	24	24			
01	L48315	166	98	96	196	8930						24	22	22			
01	M54151	170	96	130	110	7250						2	10	8	10		
01	M55384	171	96	98	293	15860						1	1	1			
01	N75124	172	134	89	200	23058								2			
01	P11866	176	97	76	214	8910						2	16	21	16		

01	Q17606	178	98	92	192	4490		3	12	17	4
01	Q17880	179	98	127	192	9230		6	8	8	8
01	S03225	183	98	75	199	13150			1	1	
01	S11273	186	77	103	276	27410					
01	S11616	187	77	81	225	21340			3	3	
01	S12164	188	88	47	139	3360			7	8	1
01	S12438	190	96	92	295	14710			2	2	
01	S56256	191	124	107	370	31390			12	12	
01	S56941	192	100	82	290	12675	18		9	3	18
01	S70661	196	120	108	515	31679					
01	S71613	197	98	132	284	8330			98	19	
01	S72846	198	98	107	374	12470			60	60	
01	S73120	199	95	95	248	6424			2	2	
01	S73531	200	98	134	276	7570			6	12	
01	S74216	201	99	133	325	8620				4	
01	S74353	202	99	133	373	13150			1	1	
01	S74832	203	95	132	316	16580			8	20	
01	S75038	204	96	129	276	7190			26	27	
01	S75175	205	98	142	346	15110			34	57	
01	T10138	206	85	82	172	5680			29	33	
01	T10412	207	96	130	322	17495				1	
01	T10549	208	98	129	349	29920			4	5	
01	T21235	210	97	134	321	20420			3	8	
01	T30414	212	98	130	265	17300			6	6	
01	T40634	213	36	27	224	3320			2	4	
01	T40771	214	68	42	314	10350					
01	U59579	217	67	55	268	17230			1	1	
01	V08844	219	98	136	346	25060			1	1	1
01	V57092	223	62	68	108	1500		290		11	4
01	V57504	224	87	83	173	5320			43	64	72
01	W00769	225	100	108	250	17500			1	1	1
01	W09557	231	67	58	122	6000		4	2	29	21
01	W09720	232	124	123	302	48770			42	42	42
01	W09927	233	100	21	201	16000			8	8	8
01	W04536	234	94	45	213	4860			57	75	72
01	W06907	235	98	55	333	11280			9	11	14
01	X38961	236	84	65	226	7480			67	29	143
01	X39933	238	86	60	211	6020			12	15	9
01	X40221	240	99	86	315	20912			4	7	3
01	X41105	242	98	89	378	24237					20
01	X41242	244	98	89	396	25600					24
01	X41310	245	96	83	279	16778			44	9	48
01	X41327	246	96	83	299	18328			9		9
01	X41615	247	109	99	382	24330			37	26	37
01	X43297	249	96	83	261	15213		15	20	20	19
01	X43434	250	96	83	273	15770		27	2	4	2
01	X43738	251									

01	YA0025	269	96	100	169	3800
01	YA0035	272	97	134	321	21020
01	Z90492	276	100	55	280	10000
01	P39172	471	108	105	218	21200
01	J36109	486	34	56	99	2940
01	J37215	488	96	85	171	7233
01	J37342	489	37	59	88	4820
01	J46252	494	77	77	147	2100
01	J47480	499	84	65	176	3790
01	L33810	510	98	122	317	14740
01	L36729	512	253	177	884	131734
01	L67234	515	124	99	425	21520
01	L76556	517	102	105	300	24400
01	I76752	519	74	84	148	5400
01	R65133	552	50	39	159	7888
01	S70116	556	97	73	418	14260
01	T10136	558	86	81	224	7325
01	T11732	559	85	82	190	5200
01	U58875	581	145	99	285	19500
01	U58878	582	360	27	359	14000
01	U58881	583	144	115	510	11905
01	X23227	608	70	48	102	1000
01	X41653	614	100	90	382	25079
01	X53846	620	95	112	275	16300
01	X60694	621	115	111	290	30082

37	10	53	113
		1	1
		12	12
11	2	2	
2	1	14	25 48
		6	16 8
		45	44
114	153	119	
29	97		
		1	
	19	19	
	350	350	
2	39	64	52
	12	12	
	4	2	
	1	1	
	2	4	
	30	96	19
	12	24	
	24	48	
	36	72	
108	13	47	75 6 18
		18	18
	3	3	
		4	

02 1 1

03	RIGGS	11	0	31	51	-106	23
03	CAMPRI	12	0	36	40	-87	29
03	HOOD	13	0	31	64	-97	50
03	DOVER	14	0	30	68	-75	08
03	SCUSE	15	0	53	19	-60	26
03	HICKAM	16	0	21	20	-157	55
03	PRANKP	17	0	50	02	8	34

04 16 17

05 17 11 12 13 16

06	C-5A	50	228	162	1453	73	428	3350	5750	6900	215400	105000	2.0
06	C-141A	141	123	109	840	120	422	3920	5750	6350	67600	30000	2.0
06	C-141C	142	1	1	1	120	432	3920	5750	6120	57200	19600	2.0 1
06	14												

07	1	12	1	4	-1
07	2	12	10	4	+1
07	3	16	1	1	-1
07	4	16	10	1	+1
07	5	13	1	2	-1
07	6	13	10	2	+1
07	7	11	1	11	-1

08	1 2			1	4	-1
08	1 2			10	4	+1

09	12	1	1	50	30	0	.5
09	12	1	2	141	100	10	.5
09	12	2	2	142	100	10	.5

10 3 4 5 6 1 1 -1

12 3 4 5 6
19 3 4 5 6
10 3 4 5 6

12 7 7

11
/

10 1 +1
1 2 -1
10 2 +1

1 11 -1

Appendix B

OUTPUTS

0 ATLANTIC DEPLOYMENT FOR NATC

CLEARANCE REQUIREMENTS -- WIDTH - 4.0 HEIGHT - 0.0 LENGTH - 0.0

VEHICLE DATA

NAME	NUMBER	WIDTH	HEIGHT	LENGTH	WEIGHT	UN. 1	UN. 2	UN. 3	UN. 4	UN. 5	UN. 6	UN. 7	UN. 8	UN. 9	UN. 10	UN. 11
REN	1	30.0	156.0	20.0	240.0	1669	1679	1649	1413	1773	1000	1000	1000	1000	1000	325
K31795	4	117.0	156.0	697.0	6161.0	37	14	14	88	287	96	28	23	0	4	5
C20414	5	77.0	66.0	254.0	30946.0	8	18	14	0	0	0	0	12	0	4	3
P39241	6	121.0	135.0	268.0	36345.0	4	4	4	0	0	12	18	17	0	1	0
P39378	7	133.0	149.0	577.0	60800.0	3	3	3	3	1	39	54	40	0	1	0
I43604	8	148.0	112.0	340.0	67700.0	6	16	12	0	0	3	0	3	0	4	0
W76816	9	145.0	131.0	233.0	48960.0	9	8	7	0	0	80	110	78	0	0	0
X48914	10	166.0	123.0	269.0	22900.0	6	6	6	1	0	41	82	36	0	0	0
X90186	11	103.0	123.0	269.0	16800.0	12	62	24	0	0	0	0	0	0	0	0
S70243	12	101.0	121.0	593.0	17435.0	1	1	1	0	0	3	5	0	0	0	0
S70517	13	119.0	67.0	416.0	16285.0	12	10	10	0	0	88	47	46	0	3	0
S72024	14	112.0	96.0	352.0	13980.0	38	37	37	3	1	86	48	98	2	14	0
S72983	15	111.0	109.0	377.0	15470.0	18	30	25	0	0	52	161	108	0	9	0
S73394	16	132.0	94.0	479.0	45870.0	2	6	6	0	0	4	4	2	0	2	0
S74079	17	103.0	146.0	347.0	15850.0	9	9	9	7	0	2	2	2	0	0	0
S74490	18	101.0	134.0	320.0	16343.0	9	10	9	8	4	11	6	5	0	0	0
X40099	19	100.0	81.0	265.0	15180.0	453	473	437	122	90	697	691	624	9	91	82
X40146	20	100.0	81.0	279.0	15570.0	392	247	238	70	29	271	259	68	0	50	0
X40283	21	100.0	81.0	329.0	14876.0	4	4	4	1	1	6	6	6	0	0	0
X40831	22	102.0	86.0	304.0	20910.0	42	179	159	0	0	32	35	26	0	48	40
X40968	23	119.0	87.0	320.0	21771.0	93	161	155	0	0	4	8	6	0	37	0
X43845	24	102.0	91.0	289.0	23847.0	45	22	22	0	0	57	24	52	0	4	0
X59326	25	102.0	86.0	264.0	19387.0	80	93	85	18	1	308	310	367	0	27	0
X59463	26	106.0	86.0	280.0	20177.0	1	1	1	0	0	1	1	5	0	0	0
X59600	27	119.0	111.0	288.0	30612.0	2	6	6	0	0	0	2	6	0	2	0
X59874	28	119.0	111.0	295.0	30082.0	14	10	10	0	0	37	51	35	0	3	0
X60696	29	121.0	99.0	360.0	33874.0	1	1	1	0	0	6	7	9	0	0	0
X62340	30	104.0	130.0	265.0	15760.0	79	72	74	2	0	35	31	38	0	16	3
X62477	31	104.0	130.0	279.0	16102.0	8	8	8	0	0	7	7	4	0	0	0
X63299	32	103.0	106.0	355.0	34820.0	46	41	48	0	0	38	44	36	0	4	10
X67486	33	102.0	98.0	391.0	26560.0	8	8	8	0	0	0	0	0	0	0	0
I45534	34	124.0	148.0	513.0	42680.0	4	4	4	0	0	0	0	0	0	0	0
A93125	35	115.0	98.0	249.0	29960.0	27	27	27	0	0	30	6	27	0	0	0
E56577	36	151.0	130.0	354.0	110560.0	3	8	8	0	0	0	0	0	0	0	0
J96694	37	110.0	105.0	192.0	24200.0	24	24	24	0	0	0	0	0	0	0	0
K50981	38	128.0	108.0	265.0	57633.0	4	12	12	0	0	0	12	12	0	0	0
K57666	39	129.0	110.0	355.0	49048.0	54	54	54	0	0	0	18	18	0	0	18
F50544	40	128.0	115.0	254.0	46448.0	18	13	61	0	0	6	4	9	0	7	15
E50681	41	139.0	116.0	321.0	107600.0	10	37	27	0	0	6	5	4	0	11	3
V13101	42	146.0	129.0	325.0	97000.0	54	324	216	0	0	17	17	17	0	117	0
V10275	43	101.0	129.0	32.0	20000.0	1	1	3	0	0	4	9	8	0	0	0
T13152	44	100.0	119.0	335.0	32780.0	1	1	1	1	1	3	5	5	0	0	0
F43077	46	106.0	89.0	203.0	15470.0	0	0	0	3	1	3	9	9	0	0	0
J74920	47	95.0	90.0	309.0	20940.0	0	0	0	4	0	9	9	9	0	0	0
X63162	48	100.0	101.0	303.0	23960.0	0	0	0	11	3	0	1	1	0	0	0
S12575	50	124.0	54.0	162.0	6540.0	0	0	0	0	2	24	24	28	0	0	0
W76268	51	90.0	84.0	155.0	20290.0	0	0	0	0	4	96	96	96	0	0	0
J35698	53	101.0	118.0	313.0	26432.0	0	0	0	0	0	0	8	7	0	0	0
K30378	54	146.0	141.0	611.0	17660.0	0	0	0	0	96	129	66	4	0	0	0
D12087	55	104.0	84.0	192.0	20125.0	89	358	397	0	0	43	0	34	0	79	26
D11401	56	79.0	91.0	168.0	13503.0	74	179	179	0	0	67	7	50	0	42	51
E11538	57	105.0	104.0	192.0	22415.0	22	138	129	0	0	15	0	25	0	29	10

E02670	58	101.0	51.0	188.0	270.0	4	6	4	8	0	24	20	29	0	0	0
E94030	59	99.0	58.0	148.0	2750.0	102	105	105	6	0	17	29	7	0	0	0
E95263	61	93.0	51.0	150.0	1600.0	2	2	2	0	0	18	18	3	0	0	0
E95537	63	78.0	50.0	147.0	1350.0	549	323	330	505	351	210	261	208	0	0	0
E95811	64	87.0	55.0	166.0	2670.0	638	659	632	79	76	647	567	519	0	0	0
E98825	65	86.0	81.0	162.0	2530.0	120	116	119	79	26	164	198	103	0	0	0
E38639	66	70.0	53.0	142.0	2895.0	58	37	34	1	0	46	38	4	0	3	3
E38776	67	87.0	91.0	200.0	6900.0	42	41	41	484	259	46	4	43	0	1	1
E39735	68	80.0	65.0	186.0	5660.0	687	469	466	484	259	273	624	332	0	85	62
E39872	69	89.0	65.0	191.0	5836.0	93	191	125	192	147	55	50	40	0	31	0
E57271	70	100.0	92.0	264.0	14600.0	21	20	18	2	0	33	37	30	0	5	0
E57403	71	100.0	92.0	278.0	15020.0	2	2	2	0	0	14	15	15	0	0	21
E59052	72	98.0	82.0	229.0	11730.0	2	2	2	0	0	11	17	19	0	0	0
E59189	73	98.0	82.0	243.0	12143.0	1	1	1	0	0	0	0	0	0	1	0
E60833	74	68.0	53.0	132.0	2450.0	1061	905	914	818	670	573	772	237	0	222	112
E57529	76	91.0	70.0	284.0	4760.0	54	0	0	0	0	0	0	0	0	0	0
E12826	82	130.0	109.0	277.0	46580.0	0	0	0	0	0	0	0	0	0	0	0
E12963	83	147.0	117.0	292.0	99500.0	0	0	0	0	0	0	0	0	0	0	51
E12482	85	46.0	42.0	225.0	4540.0	0	0	0	0	0	1	1	1	0	0	0
E18373	87	101.0	86.0	196.0	7510.0	0	0	0	0	0	1	16	10	0	0	0
E63711	90	122.0	44.0	154.0	5110.0	0	0	0	0	0	1	1	3	0	0	0
E30997	108	50.0	54.0	73.0	1675.0	0	0	0	0	0	58	155	11	0	0	0
E29660	111	132.0	150.0	626.0	7475.0	15	9	9	0	0	0	0	0	0	0	0
E11049	113	104.0	77.0	332.0	16774.0	0	0	0	0	0	0	42	48	0	0	0
E02533	117	102.0	47.0	188.0	3675.0	0	0	0	0	0	18	18	18	0	0	0
E40960	118	88.0	90.0	166.0	4905.0	0	0	0	0	58	17	33	2	0	0	0
E59646	119	88.0	84.0	145.0	18280.0	0	0	0	0	0	4	4	4	0	0	0
E59831	120	85.0	86.0	122.0	9355.0	0	0	0	0	0	2	2	2	0	0	0
E73352	122	82.0	87.0	163.0	10660.0	0	0	0	0	0	4	4	6	0	0	0
E84531	123	72.0	64.0	105.0	3260.0	0	0	0	0	0	26	37	14	0	0	0
E06424	124	124.0	111.0	434.0	9800.0	0	0	0	0	0	4	4	4	0	0	0
E40474	125	140.0	153.0	222.0	84330.0	0	0	0	0	0	4	4	4	0	0	0
E43364	126	128.0	122.0	182.0	42630.0	0	0	0	0	0	4	4	4	0	0	0
E43414	127	48.0	34.0	198.0	3356.0	0	0	0	0	0	14	16	14	0	0	0
E43429	128	104.0	118.0	459.0	65880.0	0	0	0	0	0	0	0	1	0	0	0
E50741	129	112.0	144.0	410.0	72030.0	0	0	0	0	0	1	1	1	0	0	0
E51132	131	122.0	143.0	495.0	61450.0	0	0	0	0	0	1	1	1	0	0	0
E27844	134	104.0	122.0	340.0	24760.0	0	0	0	0	0	2	2	3	0	0	0
E28212	135	103.0	110.0	290.0	21300.0	0	0	0	0	0	6	6	7	0	0	0
E29945	136	103.0	126.0	355.0	37100.0	0	0	0	0	0	2	2	3	0	0	0
E40261	138	102.0	115.0	198.0	12820.0	0	0	0	0	0	1	1	2	0	0	0
E55186	139	99.0	115.0	199.0	7370.0	0	0	0	0	0	1	1	3	0	0	0
E01855	140	103.0	131.0	394.0	18080.0	0	0	0	0	0	9	15	13	0	0	0
E01957	141	103.0	131.0	327.0	21390.0	0	0	0	0	0	6	10	10	0	0	0
E30766	144	106.0	134.0	417.0	29630.0	0	0	0	0	0	1	0	3	0	0	0
E31451	145	104.0	134.0	461.0	57304.0	0	0	0	0	0	2	0	6	0	0	0
E42100	151	87.0	68.0	176.0	46860.0	0	0	0	0	54	81	27	47	0	0	0
E74852	153	99.0	124.0	313.0	28250.0	0	0	0	0	0	13	17	13	0	0	0
E97230	155	128.0	128.0	443.0	59205.0	0	0	0	0	0	12	12	12	0	0	0
E04697	156	29.0	24.0	137.0	7000.0	0	0	0	0	0	4	4	5	0	0	0
E25215	160	92.0	104.0	169.0	8530.0	0	0	0	0	0	6	0	2	0	0	0
E57667	163	129.0	110.0	355.0	49048.0	0	0	0	0	0	4	0	12	0	0	0
E21437	164	72.0	56.0	144.0	1830.0	0	0	0	0	0	4	5	1	0	0	0
E45808	165	100.0	92.0	117.0	4265.0	0	0	0	0	0	24	24	24	0	0	0
E48315	166	102.0	96.0	196.0	8930.0	0	0	0	0	0	24	22	22	0	0	0
E54151	170	100.0	130.0	110.0	7250.0	0	0	0	0	2	10	8	10	0	0	0
E55384	171	100.0	98.0	293.0	15660.0	0	0	0	0	0	1	1	1	0	0	0
E75124	172	138.0	89.0	200.0	23058.0	0	0	0	0	0	0	0	2	0	0	0
E11866	176	101.0	76.0	214.0	8910.0	0	0	0	0	0	16	21	16	0	0	0
E17606	178	102.0	92.0	192.0	4490.0	0	0	0	2	0	12	17	4	0	0	0
E17880	179	102.0	127.0	192.0	9230.0	0	0	0	0	6	8	8	8	0	0	0

[illegible]

VEHICLE LIST 1 INCLUDES THE FOLLOWING VEHICLES

1

BASES USED IN DEPLOYMENT

NAME	NUMBER	GROUND TIME	DEG LAT	MIN LAT	DEG LONG	MIN LONG
BIGGS*	11	0.0	31.	51.	-106.	23.
CAMPBELL	12	0.0	36.	40.	-87.	29.
HCCD	13	0.0	31.	4.	-97.	50.
DOVER	14	0.0	39.	8.	-75.	28.
GOOSE	15	0.0	53.	19.	-60.	26.
HICKAM	16	0.0	21.	20.	-157.	55.
FRANKP	17	0.0	50.	2.	8.	34.

* This is the air force base used as the aerial embarkation point for FORT BLISS indicated as the onland base on page 2 and Figure 1.

LINKS INPUT	FROM	TC	DISTANCE
	HICKAM	FRANKF	LINK DENIED

TABLE OF SHORTEST ALLOWABLE DISTANCES

[illegible]

ROUTES FOR C-5A AIRCRAFT
 COMPARTMENT IS 228.0 WIDE, 162.0 HIGH AND 1453.0 LCN3 (73. PASSENGERS)
 MAXIMUM PAYLOAD IS 215400.0 REDUCED PAYLOAD IS 105000.0 RANGES ARE 3350.0 5750.0 6900.0
 SPEED IS 428. GTFH= 2.00

MINIMUM TIME FROM FRANKF TO BIGGS IS 33.59
 ROUTE IS
 FRANKF BIGGS
 MAXIMUM-FLOW ROUTE FROM BIGGS TO FRANKF IS
 BIGGS GOOSE FRANKF
 TIME IS 33.75 PAYLOAD IS 215400.0
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO CAMPBL IS 27.62
 ROUTE IS
 FRANKF CAMPBL
 MAXIMUM-FLOW ROUTE FROM CAMPBL TO FRANKF IS
 CAMPBL GOOSE FRANKF
 TIME IS 27.86 PAYLOAD IS 215400.0
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO HOOD IS 31.97
 ROUTE IS
 FRANKF HOOD
 MAXIMUM-FLOW ROUTE FROM HOOD TO FRANKF IS
 HOOD GOOSE FRANKF
 TIME IS 31.97 PAYLOAD IS 215400.0
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO HICKAM IS 49.81
 ROUTE IS
 FRANKF GOOSE HICKAM
 MAXIMUM-FLOW ROUTE FROM HICKAM TO FRANKF IS
 HICKAM BIGGS GOOSE FRANKF
 TIME IS 53.42 PAYLOAD IS 215400.0
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ROUTES FOR C-141A AIRCRAFT
 COMPARTMENT IS 123.0 WIDE, 109.0 HIGH AND 840.0 LONG (120. PASSENGERS)
 MAXIMUM PAYLOAD IS 67600.0 REDUCED PAYLOAD IS 30000.0 RANGES ARE 3920.0 5750.0 6350.0
 SPEED IS 422. JTEH= 2.00

MINIMUM TIME FROM FRANKF TO BIGGS IS 34.07
 ROUTE IS
 FRANKF BIGGS
 MAXIMUM-FLOW ROUTE FROM BIGGS TO FRANKF IS
 BIGGS GOOSE FRANKF
 TIME IS 34.23 PAYLOAD IS 67600.0
 XX

MINIMUM TIME FROM FRANKF TO CAMEBL IS 28.21
 ROUTE IS
 FRANKF CAMEBL
 MAXIMUM-FLOW ROUTE FROM CAMEBL TO FRANKF IS
 CAMEBL GOOSE FRANKF
 TIME IS 28.25 PAYLOAD IS 67600.0
 XX

MINIMUM TIME FROM FRANKF TO HOOD IS 32.43
 ROUTE IS
 FRANKF HOOD
 MAXIMUM-FLOW ROUTE FROM HOOD TO FRANKF IS
 HOOD GOOSE FRANKF
 TIME IS 32.43 PAYLOAD IS 67600.0
 XX

MINIMUM TIME FROM FRANKF TO HICKAM IS 50.51
 ROUTE IS
 FRANKF GOOSE HICKAM
 MAXIMUM-FLOW ROUTE FROM HICKAM TO FRANKF IS
 HICKAM BIGGS GOOSE FRANKF
 TIME IS 54.18 PAYLOAD IS 67600.0
 XX

ROUTES FOR C-141C AIRCRAFT
 COMPARTMENT IS 1.0 WIDE, 1.0 HIGH AND 1.0 LONG (120. PASSENGERS)
 MAXIMUM PAYLOAD IS 57200.0 REDUCED PAYLOAD IS 19600.0 RANGES ARE 3920.0 5750.0 6120.0
 SPEED IS 432. GLEPH= 2.00
 DENIED BASES 14

MINIMUM TIME FROM FRANKF TO BIGGS IS 33.28
 ROUTE IS
 FRANKF BIGGS

MAXIMUM-FLOW ROUTE FROM BIGGS TO FRANKF IS
 BIGGS GOOSE FRANKF
 TIME IS 33.43 PAYLOAD IS 57200.0

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO CAMEBL IS 27.56
 ROUTE IS
 FRANKF CAMEBL

MAXIMUM-FLOW ROUTE FROM CAMEBL TO FRANKF IS
 CAMEBL JOOSE FRANKF
 TIME IS 27.60 PAYLOAD IS 57200.0

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO HOOD IS 31.68
 ROUTE IS
 FRANKF HOOD

MAXIMUM-FLOW ROUTE FROM HOOD TO FRANKF IS
 HOOD JOOSE FRANKF
 TIME IS 31.69 PAYLOAD IS 57200.0

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINIMUM TIME FROM FRANKF TO HICKAM IS 49.34
 ROUTE IS
 FRANKF JOOSE HICKAM

MAXIMUM-FLOW ROUTE FROM HICKAM TO FRANKF IS
 HICKAM BIGGS JOOSE FRANKF
 TIME IS 52.92 PAYLOAD IS 57200.0

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

STOCK LISTS	FLAG	BASE	QUANTITY	UNIT	LIST
	1	CAMPBL	1.00	4	-1
	2	CAMPBL	10.00	4	1
	3	HICKAM	1.00	1	-1
	4	HICKAM	10.00	1	1
	5	HOOD	1.00	2	-1
	6	HOOD	10.00	2	1
	7	BIGGS	1.00	11	-1

AIRCRAFT ARRIVALS

BASE	FLIGHTS	AIRCRAFT TYPE	QUANTITY	TIME BEFORE FIRST	TIME BETWEEN
CAMPBL	1	2	C-141A	10.00	0.50
AIR CRAFT	ONLOAD BASE	TOTAL WEIGHT LOADED	UNUSED CAPACITY	CARGO WEIGHT	NUMBER OF PERSONNEL
1	C-141A	CAMPBL	31900.0	35700.0	0
2	C-141A	CAMPBL	33280.0	34320.0	0
3	C-141A	CAMPBL	45540.0	22060.0	0
4	C-141A	CAMPBL	45540.0	22060.0	0
5	C-141A	CAMPBL	45540.0	22060.0	0
6	C-141A	CAMPBL	45540.0	22060.0	0
7	C-141A	CAMPBL	45540.0	22060.0	0
8	C-141A	CAMPBL	45540.0	22060.0	0
9	C-141A	CAMPBL	45540.0	22060.0	0
10	C-141A	CAMPBL	45540.0	22060.0	0
11	C-141A	CAMPBL	45540.0	22060.0	0
12	C-141A	CAMPBL	46710.0	20890.0	0
13	C-141A	CAMPBL	46710.0	20890.0	0
14	C-141A	CAMPBL	46710.0	20890.0	0
15	C-141A	CAMPBL	46710.0	20890.0	0
16	C-141A	CAMPBL	46710.0	20890.0	0
17	C-141A	CAMPBL	46710.0	20890.0	0
18	C-141A	CAMPBL	46710.0	20890.0	0
19	C-141A	CAMPBL	46710.0	20890.0	0
20	C-141A	CAMPBL	46710.0	20890.0	0
21	C-141A	CAMPBL	46710.0	20890.0	0
22	C-141A	CAMPBL	46710.0	20890.0	0
23	C-141A	CAMPBL	46710.0	20890.0	0
24	C-141A	CAMPBL	46710.0	20890.0	0
25	C-141A	CAMPBL	46710.0	20890.0	0
26	C-141A	CAMPBL	46710.0	20890.0	0
27	C-141A	CAMPBL	46710.0	20890.0	0
28	C-141A	CAMPBL	46710.0	20890.0	0
29	C-141A	CAMPBL	46710.0	20890.0	0
30	C-141A	CAMPBL	46710.0	20890.0	0
31	C-141A	CAMPBL	46710.0	20890.0	0
32	C-141A	CAMPBL	46710.0	20890.0	0
33	C-141A	CAMPBL	46710.0	20890.0	0
34	C-141A	CAMPBL	46710.0	20890.0	0
35	C-141A	CAMPBL	33196.0	34404.0	0
36	C-141A	CAMPBL	50670.0	16930.0	0
37	C-141A	CAMPBL	50670.0	16930.0	0
38	C-141A	CAMPBL	50670.0	16930.0	0
39	C-141A	CAMPBL	50670.0	16930.0	0
40	C-141A	CAMPBL	50590.0	17010.0	0
41	C-141A	CAMPBL	53160.0	14440.0	0
42	C-141A	CAMPBL	47310.0	20290.0	0
43	C-141A	CAMPBL	47310.0	20290.0	0
44	C-141A	CAMPBL	47310.0	20290.0	0
45	C-141A	CAMPBL	47310.0	20290.0	0
46	C-141A	CAMPBL	47310.0	20290.0	0
47	C-141A	CAMPBL	47310.0	20290.0	0
48	C-141A	CAMPBL	47310.0	20290.0	0
49	C-141A	CAMPBL	47310.0	20290.0	0
50	C-141A	CAMPBL	47310.0	20290.0	0

51	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
52	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
53	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
54	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
55	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
56	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
57	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
58	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
59	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
60	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
61	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
62	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
63	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
64	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
65	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
66	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
67	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
68	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
69	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
70	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
71	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
72	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
73	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
74	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
75	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
76	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
77	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
78	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
79	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
80	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
81	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
82	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
83	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
84	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
85	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
86	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
87	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
88	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
89	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
90	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
91	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
92	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
93	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
94	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
95	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
96	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
97	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
98	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
99	C-141A	CAMPBL	13350.0	54250.0	13350.0	0
100	C-141A	CAMPBL	13350.0	54250.0	13350.0	0

ALL AIRCRAFT LOADED FROM LIST WITH FLAG NUMBER 1

AIRCRAFT ARRIVALS

BASE	FLIGHTS	AIRCRAFT TYPE	QUANTITY	TIME BEFORE FIRST	TIME BETWEEN	NUMBER OF PERSONNEL
CAMPBL	2	2	C-141C	100	10.00	0.50
1	C-141C	CAMPBL	28800.0	28400.0	0.0	120
2	C-141C	CAMPBL	28800.0	28400.0	0.0	120
3	C-141C	CAMPBL	28800.0	28400.0	0.0	120
4	C-141C	CAMPBL	28800.0	28400.0	0.0	120
5	C-141C	CAMPBL	28800.0	28400.0	0.0	120
6	C-141C	CAMPBL	28800.0	28400.0	0.0	120
7	C-141C	CAMPBL	28800.0	28400.0	0.0	120
8	C-141C	CAMPBL	28800.0	28400.0	0.0	120
9	C-141C	CAMPBL	28800.0	28400.0	0.0	120
10	C-141C	CAMPBL	28800.0	28400.0	0.0	120
11	C-141C	CAMPBL	28800.0	28400.0	0.0	120
12	C-141C	CAMPBL	28800.0	28400.0	0.0	120
13	C-141C	CAMPBL	28800.0	28400.0	0.0	120
14	C-141C	CAMPBL	28800.0	28400.0	0.0	120
15	C-141C	CAMPBL	28800.0	28400.0	0.0	120
16	C-141C	CAMPBL	28800.0	28400.0	0.0	120
17	C-141C	CAMPBL	28800.0	28400.0	0.0	120
18	C-141C	CAMPBL	28800.0	28400.0	0.0	120
19	C-141C	CAMPBL	28800.0	28400.0	0.0	120
20	C-141C	CAMPBL	28800.0	28400.0	0.0	120
21	C-141C	CAMPBL	28800.0	28400.0	0.0	120
22	C-141C	CAMPBL	28800.0	28400.0	0.0	120
23	C-141C	CAMPBL	28800.0	28400.0	0.0	120
24	C-141C	CAMPBL	28800.0	28400.0	0.0	120
25	C-141C	CAMPBL	28800.0	28400.0	0.0	120
26	C-141C	CAMPBL	28800.0	28400.0	0.0	120
27	C-141C	CAMPBL	28800.0	28400.0	0.0	120
28	C-141C	CAMPBL	28800.0	28400.0	0.0	120
29	C-141C	CAMPBL	28800.0	28400.0	0.0	120
30	C-141C	CAMPBL	28800.0	28400.0	0.0	120
31	C-141C	CAMPBL	28800.0	28400.0	0.0	120
32	C-141C	CAMPBL	28800.0	28400.0	0.0	120
33	C-141C	CAMPBL	28800.0	28400.0	0.0	120
34	C-141C	CAMPBL	28800.0	28400.0	0.0	120
35	C-141C	CAMPBL	28800.0	28400.0	0.0	120
36	C-141C	CAMPBL	28800.0	28400.0	0.0	120
37	C-141C	CAMPBL	28800.0	28400.0	0.0	120
38	C-141C	CAMPBL	28800.0	28400.0	0.0	120
39	C-141C	CAMPBL	28800.0	28400.0	0.0	120
40	C-141C	CAMPBL	28800.0	28400.0	0.0	120
41	C-141C	CAMPBL	28800.0	28400.0	0.0	120
42	C-141C	CAMPBL	28800.0	28400.0	0.0	120
43	C-141C	CAMPBL	28800.0	28400.0	0.0	120
44	C-141C	CAMPBL	28800.0	28400.0	0.0	120
45	C-141C	CAMPBL	28800.0	28400.0	0.0	120
46	C-141C	CAMPBL	28800.0	28400.0	0.0	120
47	C-141C	CAMPBL	28800.0	28400.0	0.0	120
48	C-141C	CAMPBL	28800.0	28400.0	0.0	120
49	C-141C	CAMPBL	28800.0	28400.0	0.0	120
50	C-141C	CAMPBL	28800.0	28400.0	0.0	120

51	C-141C	CAMPBL	28800.0	28400.0	0.0	120
52	C-141C	CAMPBL	28800.0	28400.0	0.0	120
53	C-141C	CAMPBL	28800.0	28400.0	0.0	120
54	C-141C	CAMPBL	28800.0	28400.0	0.0	120
55	C-141C	CAMPBL	28800.0	28400.0	0.0	120
56	C-141C	CAMPBL	28800.0	28400.0	0.0	120
57	C-141C	CAMPBL	28800.0	28400.0	0.0	120
58	C-141C	CAMPBL	28800.0	28400.0	0.0	120
59	C-141C	CAMPBL	28800.0	28400.0	0.0	120
60	C-141C	CAMPBL	28800.0	28400.0	0.0	120
61	C-141C	CAMPBL	28800.0	28400.0	0.0	120
62	C-141C	CAMPBL	28800.0	28400.0	0.0	120
63	C-141C	CAMPBL	28800.0	28400.0	0.0	120
64	C-141C	CAMPBL	28800.0	28400.0	0.0	120
65	C-141C	CAMPBL	28800.0	28400.0	0.0	120
66	C-141C	CAMPBL	28800.0	28400.0	0.0	120
67	C-141C	CAMPBL	28800.0	28400.0	0.0	120
68	C-141C	CAMPBL	28800.0	28400.0	0.0	120
69	C-141C	CAMPBL	28800.0	28400.0	0.0	120
70	C-141C	CAMPBL	28800.0	28400.0	0.0	120
71	C-141C	CAMPBL	28800.0	28400.0	0.0	120
72	C-141C	CAMPBL	28800.0	28400.0	0.0	120
73	C-141C	CAMPBL	28800.0	28400.0	0.0	120
74	C-141C	CAMPBL	28800.0	28400.0	0.0	120
75	C-141C	CAMPBL	28800.0	28400.0	0.0	120
76	C-141C	CAMPBL	28800.0	28400.0	0.0	120
77	C-141C	CAMPBL	28800.0	28400.0	0.0	120
78	C-141C	CAMPBL	28800.0	28400.0	0.0	120
79	C-141C	CAMPBL	28800.0	28400.0	0.0	120
80	C-141C	CAMPBL	28800.0	28400.0	0.0	120
81	C-141C	CAMPBL	28800.0	28400.0	0.0	120
82	C-141C	CAMPBL	28800.0	28400.0	0.0	120
83	C-141C	CAMPBL	28800.0	28400.0	0.0	120
84	C-141C	CAMPBL	28800.0	28400.0	0.0	120
85	C-141C	CAMPBL	28800.0	28400.0	0.0	120
86	C-141C	CAMPBL	28800.0	28400.0	0.0	120
87	C-141C	CAMPBL	28800.0	28400.0	0.0	120
88	C-141C	CAMPBL	28800.0	28400.0	0.0	120
89	C-141C	CAMPBL	28800.0	28400.0	0.0	120
90	C-141C	CAMPBL	28800.0	28400.0	0.0	120
91	C-141C	CAMPBL	28800.0	28400.0	0.0	120
92	C-141C	CAMPBL	28800.0	28400.0	0.0	120
93	C-141C	CAMPBL	28800.0	28400.0	0.0	120
94	C-141C	CAMPBL	28800.0	28400.0	0.0	120
95	C-141C	CAMPBL	28800.0	28400.0	0.0	120
96	C-141C	CAMPBL	28800.0	28400.0	0.0	120
97	C-141C	CAMPBL	28800.0	28400.0	0.0	120
98	C-141C	CAMPBL	28800.0	28400.0	0.0	120
99	C-141C	CAMPBL	28800.0	28400.0	0.0	120
100	C-141C	CAMPBL	28800.0	28400.0	0.0	120

ALL AIRCRAFT LOADED FROM LIST WITH FLAG NUMBER

DELIVERIES OF MATERIEL AND PERSONNEL FROM FREIGHTY GROUPE 1 TO OFFLOAD BASE

PERSONNEL	PERCENT OF PERSONNEL	PERCENT OF CARGO	DAY HOUR	
2520	17.83	24.10	2	0 1
1284C	90.87	53.36	4	0 1
14130	100.00	75.65	6	0 1
14130	100.00	92.31	8	0 1
14130	100.00	100.00	10	0 1

ITEMS IN PRIORITY GROUP NOT DELIVERED

VEHICLES IN PRIORITY GROUP

NAME	NUMBER	WIDTH	HEIGHT	LENGTH	WEIGHT	QUANTITY
CARGO WEIGHT IS		0.0	TONS	NUMBER OF PERSONNEL IS		0
CARGO COVERS		0.0	THOUSAND SQUARE FEET OF FLOOR SPACE, AND WEIGHTS			0.0 POUNDS PER SQUARE FOOT

[illegible]

X38639	66	70.0	53.0	142.0	2895.0	95
X38776	67	87.0	91.0	200.0	6900.0	83
X39735	68	80.0	65.0	186.0	5660.0	1156
X39872	69	80.0	65.0	191.0	5836.0	284
X57271	70	100.0	92.0	264.0	14600.0	41
X57408	71	100.0	92.0	278.0	15022.0	4
X59052	72	98.0	82.0	229.0	11703.0	4
X59189	73	98.0	82.0	243.0	12143.0	2
X60833	74	68.0	53.0	132.0	2450.0	1966
X57529	76	91.0	70.0	284.0	4780.0	54
X29660	111	132.0	150.0	626.0	7475.0	24
S56941	192	104.0	82.0	290.0	12675.0	18
CARGO WEIGHT IS		73262.94 TONS		NUMBER OF PERSONNEL IS		33480
CARGO COVERS		1533.5 THOUSAND SQUARE FEET OF FLOOR SPACE, AND WEIGHS		95.55 POUNDS PER SQUARE FOOT		

DELIVERIES OF MATERIEL AND PERSONNEL FROM PRIORITY GROUP 2 TO OFFLOAD BASE

PERSONNEL	PERCENT OF PERSONNEL	PERCENT OF CARGO	DAY HOUR	
8280	C.C	C.C	4	0 *
16560	24.73	C.C	6	0 *
19550	49.46	C.C	8	0 *
27950	58.39	4.81	10	0 1 *
30950	83.48	11.74	12	0 1 *
33480	92.44	17.22	14	0 1 *
33480	100.00	23.13	16	0 1 *
33480	100.00	29.85	18	0 1 *
33480	100.00	36.55	20	0 1 *
33480	100.00	41.08	22	0 1 *
33480	100.00	45.96	24	0 1 *
33480	100.00	52.46	26	0 1 *
33480	100.00	58.42	28	0 1 *
33480	100.00	62.14	30	0 1 *
33480	100.00	64.06	32	0 1 *
33480	100.00	66.10	34	0 1 *
33480	100.00	71.57	36	0 1 *
33480	100.00	72.41	38	0 1 *
33480	100.00	79.06	40	0 1 *
33480	100.00	80.42	42	0 1 *
33480	100.00	84.71	44	0 1 *
33480	100.00	86.35	46	0 1 *
33480	100.00	90.62	48	0 1 *
33480	100.00	92.67	50	0 1 *
33480	100.00	94.16	52	0 1 *
33480	100.00	95.23	54	0 1 *
33480	100.00	98.14	56	0 1 *
33480	100.00	98.59	58	0 1 *
33480	100.00	100.00	60	0 1 *

ITEMS IN PRIORITY GROUP NOT DELIVERED

VEHICLES IN PRIORITY GROUP

NAME	NUMBER	WIDTH	HEIGHT	LENGTH	WEIGHT	QUANTITY
CARGO WEIGHT IS			0.0 TONS	NUMBER OF PERSONNEL IS	0	
CARGO COVERS			0.0 THOUSAND SQUARE FEET OF FLOOR SPACE, AND WEIGHTS		0.0 POUNDS PER SQUARE FOOT	

PRIORITY GROUP 3
PLANS

QUANTITY	UNIT	LIST
11	11	-1

VEHICLES IN PRIORITY GROUP

NAME	NUMBER	WIDTH	HEIGHT	LENGTH	WEIGHT	QUANTITY
K31795	4	117.0	156.0	997.0	6161.0	5
C20414	5	77.0	66.0	264.0	30946.0	3
X40039	19	100.0	81.0	265.0	15180.0	82
X40831	22	102.0	86.0	304.0	20910.0	40
X62340	30	104.0	130.0	265.0	15760.0	3
X63299	32	103.0	106.0	355.0	34820.0	10
X57666	39	129.0	110.0	355.0	49048.0	18
R50544	40	128.0	115.0	254.0	48448.0	15
R50681	41	139.0	118.0	321.0	107600.0	3
D12087	55	104.0	84.0	192.0	20125.0	26
D11401	56	79.0	91.0	168.0	13503.0	51
D11538	57	105.0	104.0	192.0	22415.0	10
X38639	66	76.0	53.0	142.0	2895.0	3
X38776	67	87.0	91.0	200.0	6900.0	1
X39735	68	80.0	65.0	186.0	5660.0	62
X57408	71	100.0	92.0	278.0	15022.0	21
X60833	74	68.0	53.0	132.0	2450.0	112
V12826	82	130.0	109.0	277.0	46580.0	51
V12963	83	147.0	117.0	292.0	99500.0	51

CARGO WEIGHT IS 7187.58 TONS NUMBER OF PERSONNEL IS 0

CARGO COVERS 95.5 THOUSAND SQUARE FEET OF FLOOR SPACE, AND WEIGHTS 150.46 POUNDS PER SQUARE FOOT

ITEMS IN PRIORITY GROUP NOT DELIVERED

VEHICLES IN PRIORITY GROUP

NAME	NUMBER	WIDTH	HEIGHT	LENGTH	WEIGHT	QUANTITY
CARGO WEIGHT IS						0
CARGO COVERS						

C.C TONS NUMBER OF PERSONNEL IS 0

C.C THOUSAND SQUARE FEET OF FLOOR SPACE, AND WEIGHTS

0.0 POUNDS PER SQUARE FOOT

CUMULATIVE DELIVERIES OF MATERIAL AND PERSONNEL FROM ALL PRIORITY GROUPS TO OFFLOAD BASE

PERSONNEL	PERCENT OF PERSONNEL	PERCENT OF CARGO	DAY	HCUR	
2520	5.29	2.16	2	0	1*
12840	26.97	4.79	4	0	1*
24410	47.07	6.79	6	0	1*
30690	64.46	8.29	8	0	1*
33680	70.74	12.96	10	0	1*
42080	88.38	18.71	12	0	1*
45080	94.63	23.25	14	0	1*
47610	100.00	28.15	16	0	1*
47610	100.00	33.72	18	0	1*
47610	100.00	39.27	20	0	1*
47610	100.00	43.03	22	0	1*
47610	100.00	47.07	24	0	1*
47610	100.00	52.46	26	0	1*
47610	100.00	57.40	28	0	1*
47610	100.00	60.49	30	0	1*
47610	100.00	62.08	32	0	1*
47610	100.00	63.76	34	0	1*
47610	100.00	68.30	36	0	1*
47610	100.00	69.00	38	0	1*
47610	100.00	74.51	40	0	1*
47610	100.00	75.64	42	0	1*
47610	100.00	79.19	44	0	1*
47610	100.00	80.55	46	0	1*
47610	100.00	84.09	48	0	1*
47610	100.00	85.79	50	0	1*
47610	100.00	87.03	52	0	1*
47610	100.00	87.91	54	0	1*
47610	100.00	90.32	56	0	1*
47610	100.00	90.70	58	0	1*
47610	100.00	94.40	60	0	1*
47610	100.00	97.09	62	0	1*
47610	100.00	98.64	64	0	1*
47610	100.00	100.00	66	0	1*

[illegible]

AIRCRAFT AVAILABILITY - C-5A

DAY	HOUR
61	22
61	22
61	23
62	2
62	3
62	4
62	4
62	5
62	5
62	7
62	7
62	8
62	8
62	9
62	9
62	21
63	21
64	1
64	2
64	2
64	3
64	3
64	14
64	14
64	15
64	15
64	16
64	16
64	17
64	17

AIRCRAFT AVAILABILITY - C-141A

DAY	4	4	4	4	4	5	5	5	5	6	6	6	7	7	8	8	9	9	10	10	11	11	11	12	12	12	12	13	13	13	13	14	14	14	14	14	15	15	15	15	16	16		
HOURLY	4	4	4	4	4	5	5	5	5	6	6	6	7	7	8	8	9	9	10	10	11	11	11	12	12	12	12	13	13	13	13	14	14	14	14	14	15	15	15	15	16	16		
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												
59																																												

AIRCRAFT AVAILABILITY - C-141C

DAY	HOUR
10	8
10	9
10	14
10	15
10	15
10	16
10	16
10	17
10	17
10	18
10	18
10	19
10	19
10	20
10	20
10	21
10	21
10	22
10	22
10	23
11	7
11	8
11	8
11	9
11	9
11	10
11	10
11	11
11	11
11	12
11	12
11	13
11	13
11	14
11	14
11	15
11	15
11	16
11	16
11	17
11	17
11	18
11	18
11	19
11	19
11	20
11	20
11	21
11	21
11	22
11	22
11	23
11	23
12	0
12	0
12	1
12	1
12	2

2	2
12	3
12	15
13	15
13	16
13	16
13	17
13	17
13	18
13	19
13	19
13	20
13	20
13	21
13	21
13	22
13	22
13	23
13	23
14	0
14	0
14	1
14	1
14	2
14	2
14	3
14	3
14	4
14	4
14	5
14	5
14	6
14	6
14	7
14	7
14	8
14	8
14	9
14	9
14	10
14	10

TOTAL SORTIES	
AIRCRAFT	NUMBER OF SORTIES
C-5A	615
C-141A	1882
C-141C	398

Appendix C

PROGRAMMER'S GUIDE

INTRODUCTION

This appendix is intended to be used by a programmer who is maintaining or modifying the Army Deployment Simulator computer program described in this report, although the section describing data input formats will be of use to the more casual user.

The program has a network phase and a loading phase. The loading phase is described in the greater detail.

As an aid to possible program modifications, COMMON block variables are defined. There is a table of program references to these variables. All input formats are listed with program variable names. The arguments for each subprogram are also explained.

Program modifications and problems, resolved and unresolved, are noted. In addition, there are some comments for improvement.

GENERAL STRUCTURE

The MAIN program calls subprograms to perform the tasks as illustrated in Fig. C.1.

NETWORK PHASE

Overview

The MAIN program clears COMMON, then calls a system routine to bypass printing error messages for division by zero. It then calls FRST to read data from card types 00, 01, and 02. Figure C.2 shows the calling sequence.

NETW controls the computations for the network phase. Its function is to input the data for the bases, find the minimum distances between bases, compute the minimum time back to the onload base along permissible links. It computes the routes to the offload base, using maximum flow as the criterion.

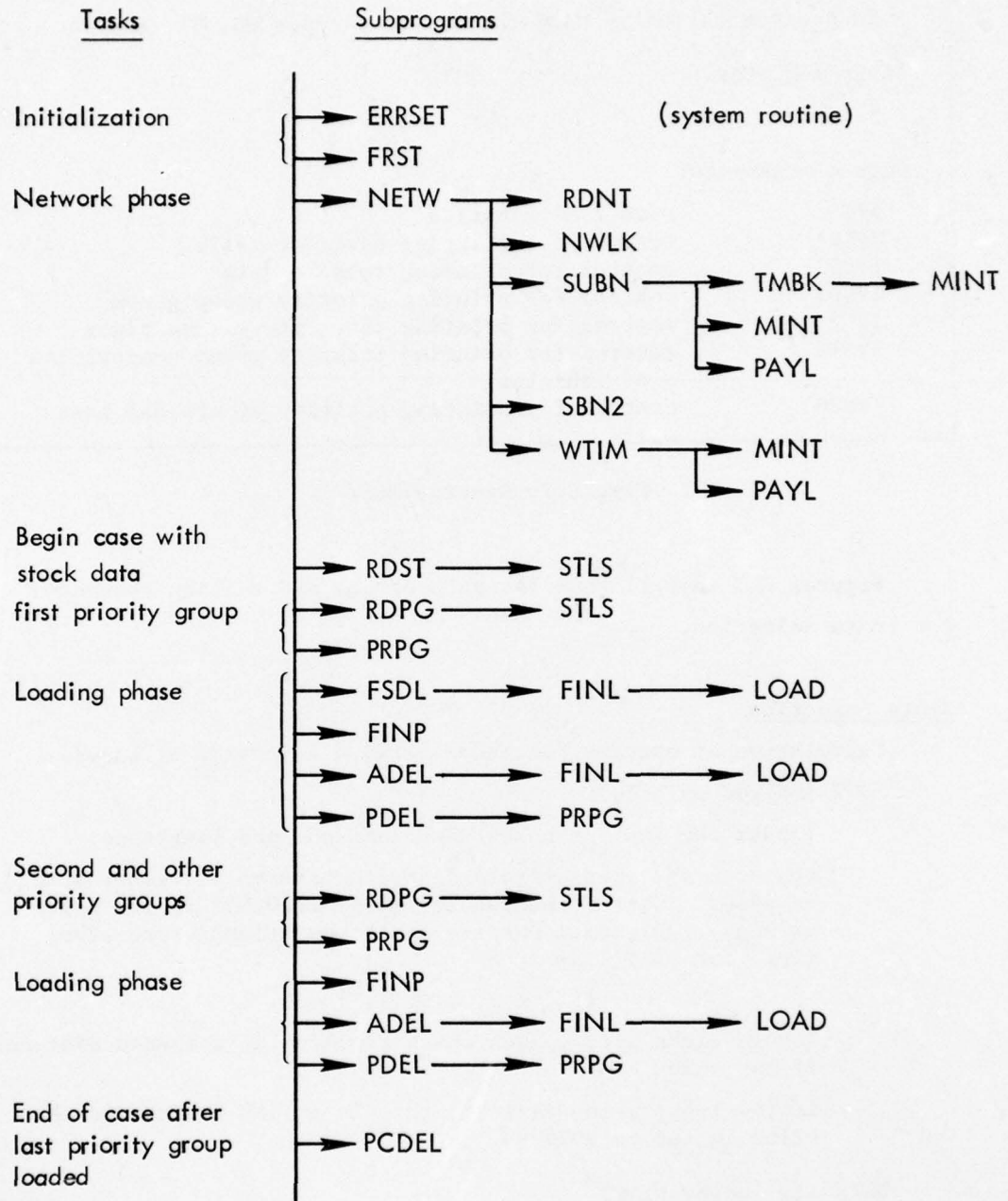


Fig. C-1

FRST (NVT, IVSP1, IVSP2, IVSP3, IVSP4, IVSP5, IVSP6)

Subprogram called by MAIN to read card types 00, 01, and 02.

Input arguments:

None

Output arguments:

NVT	number of vehicles
IVSP1	control for printing distance table
IVSP2	control for printing vehicle data
IVSP3	control for printing priority group graph
IVSP4	control for printing aircraft release times
IVSP5	control for printing priority group composition of vehicles
IVSP6	control for printing activity at offload base

Fig. C.2—Routine FRST

Figures C.3 to C.11 show the subprograms and calling sequences for route selection.

Route Selection

Calculation of nonstop distances between all pairs of bases.

RDNT (called by *NETW*)

Inputs the list of bases with latitude and longitude.

Computes the great circle distance between all combinations of bases. Stores the results in *DIST*(50,50) in the upper triangle. Diagonal entries and lower triangle are zero; i.e., for *DIST*(*I*,*J*)

if $I \leq J$, $DIST(I,J) = 0$.

Inputs links with values which replace the computed distance if the value > 0 .

If the input value is zero, this is a link denied and the value is set to 9999999.

NWLK (called by *NETW*)

Computes the shortest distance between nodes. These values are the same as those computed in *RDNT* except for links denied. Here the shortest route is computed as the sum of acceptable links from *I* to *J*, and the value 9999999 is replaced. No record is kept of the route. In later calculations, link denial information is not available.

SUBN (called by *NETW*) finds the node subnetwork

TMBK (called by SUBN) computes the time back

Reads the list of bases denied to A/C.

Creates the array NODE with a list of acceptable bases in descending order starting in position NODE(2). NODE(1) = the offload base. If there are fewer than 50 acceptable bases, it stores -1 in the slot following the last base.

MINT (called by TMBK)--The flow of this routine is described below:

MINT computes the minimum time between bases along a route where for each link in the route, the distance is \leq the given distance (here it is the ferry range).

For a given "TO" base, it computes time for each "FROM" base if the distance between bases is \leq the given distance.

Time = (base distance/speed of A/C) \times (1 + ground time per flying hour) + ground time at "TO" base + current minimum time at "FROM" base. If the time computed $<$ the time stored previously for "TO" base, store the new time, "FROM" base index, and distance. This distance is the maximum of DIST(NTO, FROM) computed in NWLK, and the current value of the distance CL(FROM) computed in MINT. After the range of all "TO" bases is computed, repeat the computation loop if any of the distance values have been replaced.

The array TMIN (NTO) contains the minimum time between TO and FROM base.

NP (NTO) contains the index of the FROM base.

CL (NTO) contains the distance along the route between bases.

The computation terminates when no new minimum time is found on the current pass.

MINT returns the following values to SUBN:

TB array = the time back from TMIN

NSAVE taken from NP in MINT

Note that the CL distance array will change with each call to MINT. Prepare to find the route out, where maximum flow is the criterion.

Clear FLOW array.

Set CLM to ferry range (maximum distance in MINT computation).

① \rightarrow Call MINT; on return, set CLMX to zero; compute flow for each node ND.

- a. compute the payload for the A/C and distance for the current node link. PAYL function computes this value.
- b. compute flow = payload/(time out and time back).
- c. store flow in FLOW(ND) if $>$ previous value. Store list of "nodes from" (NP(J) array) in NSUBN (ND,J) set CLMX to CL(ND) if CL(ND) $>$ CLMX.

If the distance CLM used for MINT \leq the range for maximum payload \rightarrow DONE; or if CLMX, current max distance, $\leq 0 \rightarrow$ DONE; else set CLM = CLMX - 1; go compute MINT and payloads \leq new range \rightarrow ①

NETW (IWAR, NACT, NNODES, IVSP1)

Subprogram called by MAIN to perform analysis.

Input arguments:

IVSP1 control for printing distance tables

Output arguments:

IWAR	offload base
NACT	number of aircraft types
NNODES	number of nodes (bases)

Fig. C.3—Routine NETW

RDNT (NNODES)

Subprogram called by NETW to read network data card types 03 and 04.

Input arguments:

None

Output argument:

NNODES number of nodes (bases)

Fig. C.4—Routine RDNT

NWLK (NNODES, IVSP1)

Subprogram called by NETW to compute minimim distance along admissible links.

Input arguments:

NNODES	number of nodes (bases)
IVSP1	control for printing distance table

Output arguments:

None

Fig. C.5—Routine NWLK

SUBN (IAC, IWAR, NNODES, NBDN)

Subprogram called by NETW for each aircraft to find node subnetwork (routes out and back).

Input arguments:

IAC	aircraft type
IWAR	offload base
NNODES	number of nodes (bases)
NBDN	number of bases denied to the aircraft

Output arguments:

None

Fig. C.6—Routine SUBN

TMBK (IWAR, SPEEDX, RANGE, NNODES, NBDN)

Subprogram called by SUBN to compute minimum time back from offload base to each onload base.

Input arguments:

IWAR	offload base
SPEEDX	speed of the aircraft
RANGE	maximum distance
NNODES	number of nodes (bases)
NBDN	number of bases denied

Output arguments:

None

Fig. C.7—Routine TMBK

MINT (CLMAX, SPEEDX, WINDF, TMINX, TGDEST, CLX)

Subprogram called by SUBN, TMBK, or WTIM to find minimum time through a network.

Input arguments:

CLMAX	maximum distance
SPEEDX	speed of aircraft
WINDF	(never referenced)

Output arguments:

TMINX	minimum time between onload and offload base route, including ground time at each link and aircraft ground time per flying hour
TGDEST	total ground time at offload base
CLX	distance for route

Fig. C.8—Routine MINT

PAYL (I, DX)

Function subprogram called by SUBN or WTIM to compute maximum payload for a given aircraft and distance combination.

Input arguments:

I	aircraft
DX	distance

Fig. C.9—Routine PAYL

SBN2 (NOL, IWAR)

Subprogram called by NETW for each onload base for each aircraft to find routes out and back.

Input arguments:

NOL	onload base
IWAR	offload base

Output arguments:

None

Fig. C.10—Routine SBN2

WTIM (IAC)

Subprogram called by NETW for each onload base for each aircraft to compute weight-time table.

Input argument:

IAC aircraft type

Output arguments:

None

Fig. C.11—Routine WTIM

Payload/Time Out Table

After the maximum flow route has been determined from each onload base, a table of payloads and time to fly over the route is prepared. Subprogram SBN2 stores the list of bases for the route in the array NODE for subprogram WTIM to use. For a given combination of aircraft, onload base, and maximum flow route, time and distance are computed over a range of distances from ferry range to maximum payload range. For each distance in the table the function PAYL is used to determine the payload that the aircraft can carry.

NETW stores the time and payload combinations for each base and aircraft. Figures C.3, C.9, C.10, and C.11 have the calling sequence for the programs.

STOCK DATA AND PRIORITY GROUPS

Stock data are input by RDST. Subprogram STLS computes the quantities. Priority group data are input by RDPG.

The calling sequences for these programs are in Figs. C.12, C.13, and C.14. Figure C.15 is for the program that prints stock data.

LOADING METHOD

Initial Sortie for Each Aircraft

The FSDL subprogram reads the type 09 control cards for aircraft arrivals. Its calling sequence is shown in Fig. C.16. It reads one

RDST (NSTL, NVT, NNODES)

Subprogram called by MAIN to read stock data from card type 07 and to create list of available stock.

Input arguments:

NVT	number of vehicles
NNODES	number of nodes (bases)

Output argument:

NSTL	number of stock lists
------	-----------------------

Fig. C.12—Routine RDST

STLS (IZ, NVT, NNEW)

Subprogram called by RDST or RDPG to store list of vehicles.

Input arguments:

IZ	stock list number
NVT	number of vehicles
NNEW	1 if new stock list 0 if continuation of stock list

Output arguments:

None

Fig. C.13—Routine STLS

RDPG (NVT, ISW, NPGC)

Subprogram called by MAIN to read data in a priority group.

Input arguments:

NVT	number of vehicles
NPGC	priority group number

Output argument:

ISW	card type
-----	-----------

Fig. C.14—Routine RDPG

PRPG (NVT, TWT, NPAXPG, IVSP5)

Subprogram called by MAIN to print priority group data.

Input arguments:

NVT	number of vehicles
IVSP5	print control for output of priority group composition

Output arguments:

TWT	cargo weight
NPAXPG	number of personnel

Fig. C.15—Routine PRPG

FSDL (NVT, NNODES, NACT, NSTL, NACF, IWAR)

Subprogram called by MAIN to find first deliveries for the aircraft.

Input arguments:

NVT	number of vehicles
NNODES	number of nodes (bases)
NACT	number of aircraft types
NSTL	number of stock lists
IWAR	offload base number

Output argument:

NACF	total number of aircraft flown
------	--------------------------------

Fig. C.16—Routine FSDL

card and verifies the data. For each aircraft in the group it calls FINL to find the load and time to fly the aircraft. Each type 09 card specifies two flag numbers. A flag number is associated with a stock list and an onload base. The flag numbers used for the initial sortie must be one of the flag numbers in the first priority group entered by card type 08. The program attempts to load the aircraft with vehicles in the list specified by the first of the two flag numbers. Note that both flag numbers must be for the same onload base. Upon return from

FINL, if the aircraft contains a load, it computes the time at which the aircraft is ready for redeployment, which is equal to:

hours before first arrival
+ (number of aircraft arrived less one) multiplied by the time between arrivals
+ time to fly load to offload base and ground time at offload base.

The time in the array ARR and the aircraft number in the array NTARR are recorded.

The program computes the time increment for the output from the time in ARR divided by the hour increment from card type 00 and adds "one" to obtain a subscript for the arrays that accumulate cargo weight, number of personnel flown for the current priority and the total of all priorities, and the number of aircraft available for departure.

It then subtracts the ground time at the offload base from the time in ARR and computes the "arrival" time increment and updates the number of aircraft that arrived. It increases the number of sorties by one.

The process is repeated for each aircraft in the current group.

Then the information for the next card type 09 is read, and the cycle repeated until there are no aircraft arriving.

If the aircraft contains no load, a check is made to determine whether the flag number is the first one specified for this aircraft. If yes, the second flag number is used to try to load this aircraft and succeeding aircraft in this group. When the aircraft is empty and the flag number is the second one, the empty aircraft is flown to the offload base. The redeployment time is computed as for a plane with a load. There will be a valid time which will be close to the time for the ferry range.

After all aircraft have been processed, the redeployment array ARR and its associated array NTARR are ordered on ascending time for redeployment.

Sorties Subsequent to Initial One for Each Aircraft

The FINP subprogram (Fig. C.17) creates a table, NPAC for each aircraft type. It shows the flag numbers in the current priority group. The flag numbers in NPAC are arranged in descending order of maximum flow that can be delivered for the aircraft-and-base combination associated with the flag.

FINP (NACT, NSTL)

Subprogram called by MAIN to create a table of priority flags for each aircraft. The table of flags is arranged in descending order of maximum flow.

Input arguments:

NACT	number of aircraft types
NSTL	number of stock lists

Output arguments:

None
The contents of the table created are in the array NPAC

Fig. C.17—Routine FINP

The ADEL subprogram (Fig. C.18) controls the allocation of aircraft for redeployment to onload bases, sets the parameters for the load routines. It maintains (1) the counts for total cargo and personnel delivered for the current priority group and (2) cumulative values for all priority groups. It counts the aircraft that arrive at the offload base in the time increment specified for output. It does the same for aircraft that depart. It updates the number of sorties. It inserts the redeployment time for the aircraft arriving at the offload base into the appropriate position of the ARR array to maintain the ascending order of time.

The program starts by selecting the first entry in NTARR array. This is the aircraft ready for deployment. The first flag is chosen for the aircraft from the NPAC table created by subprogram FINP. The onload base associated with the flag is determined and FINL is called

ADEL (NVT, NACF, IWAR)

Subprogram called by MAIN to select aircraft to load. It calls loading subprograms, then records deliveries for current priority groups.

Input arguments:

NVT	number of vehicles
NACF	total number of aircraft flown
IWAR	offload base

Output arguments:

None

Fig. C.18—Routine ADEL

to load the aircraft. If the aircraft cannot be loaded, the next flag number for the same aircraft is chosen and an attempt again is made to load. If the list of flags for this aircraft and priority group has been exhausted, the program goes to the next aircraft in the NTARR array.

After the aircraft has been loaded and the ARR and NTARR arrays have been adjusted, the next aircraft available is in the current position of the NTARR array.

ADEL continues through the NTARR array until the last aircraft cannot be loaded.

Loading the Aircraft

The FINL subprogram (Fig. C.19) controls (1) finding the load for the aircraft and (2) determining the time to fly the load. If this is the first time that the flag is specified by the input parameter LIST, the maximum payload for the aircraft and base is obtained from the maximum payload table in the array WTMV. The LOAD subprogram is called to find the load configuration for the aircraft using the current stock available for the flag in the priority group. The weight loaded will be less than or equal to the maximum payload given to LOAD. Upon return from LOAD, FINL searches the payload table to find the time

AD-A031 975

RAND CORP SANTA MONICA CALIF

F/G 15/7

THE ARMY DEPLOYMENT SIMULATOR WITH A DATA BASE OF ARMY UNITS AN--ETC(U)

SEP 76 J H HAYES, L CUTLER

F44620-73-C-0011

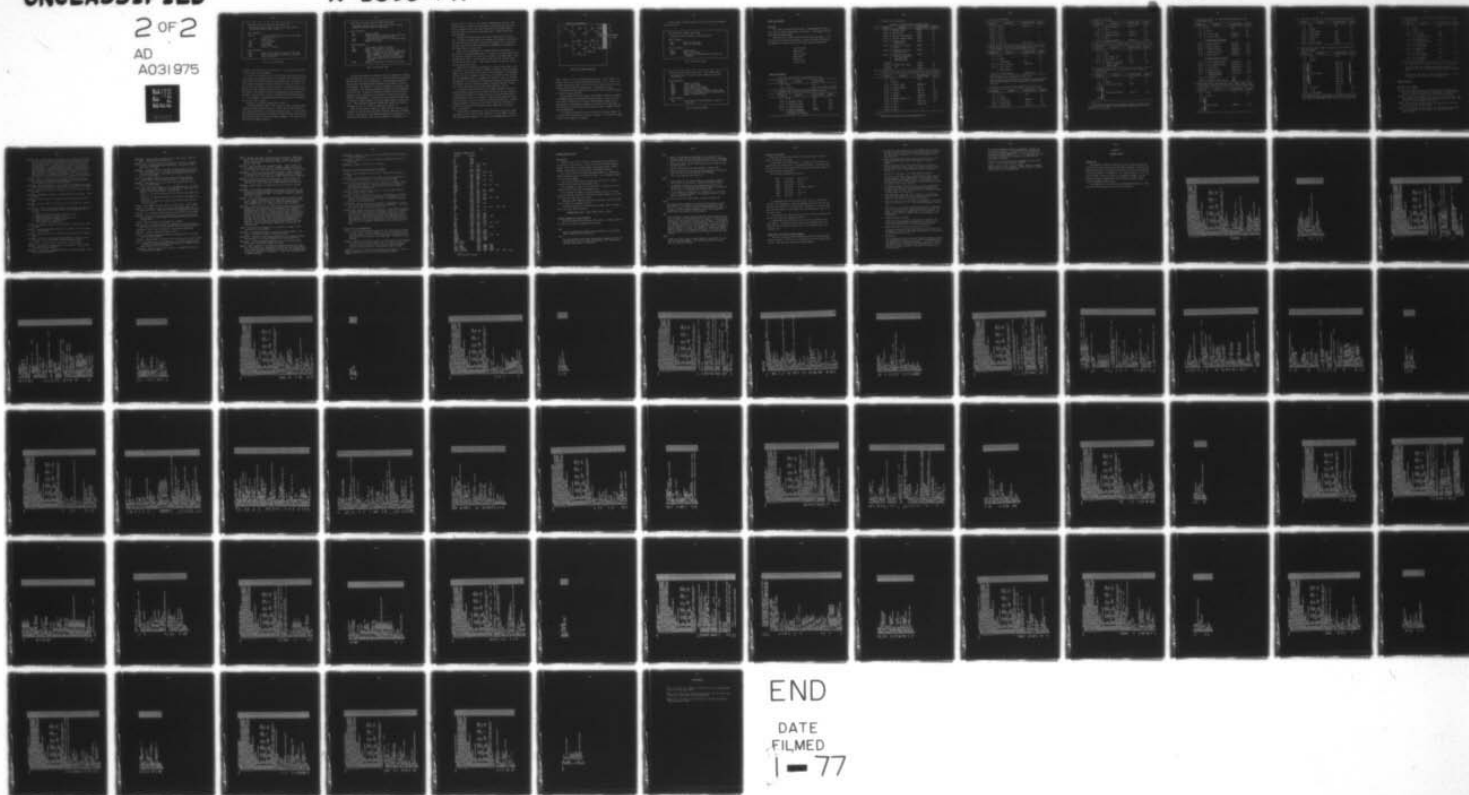
R-1893-PR

NL

UNCLASSIFIED

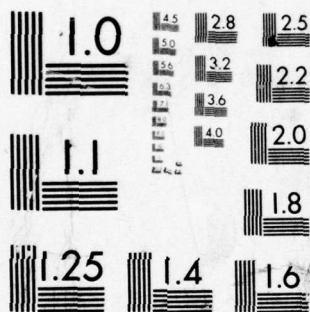
2 OF 2

AD
A031 975



END

DATE
FILMED
1-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FINL (NSW, IAC, LIST, NOB, NVT, WTC, NPAX, TTO)

Subprogram called by FSDL or ADEL to find the load for the aircraft and the time to fly it.

Input arguments:

NSW = 1	the first time loading from the flag number stored in LIST
= 0	other times
IAC	aircraft number
LIST	flag number
NOB	base number
NVT	number of vehicles

Output arguments:

WTC	total weight loaded on aircraft, from LOAD
NPAX	number of personnel on aircraft, from LOAD
TTO	time to fly load

Fig. C.19—Routine FINL

it takes to fly the load for the aircraft. It then saves the current aircraft number and flag number.

If this is not the first time that FINL has been called with the flag number, the subprogram checks to see whether the flag and the aircraft are the same as the ones on the previous load. If they are not, the situation is treated as before and a call is set up to LOAD. If the aircraft and flag are the same as before, a check is made to see whether the same configuration can be used. If so, the stock list and the count for the number of times the configuration may be used are adjusted. The time value is still available from the previous load.

If the number of configurations have been used up, the parameters are set and LOAD is called.

Return to the calling program ADEL or FSDL.

The LOAD subprogram (Fig. C.20), called by FINL, determines which vehicles to place on the aircraft at a given onload base, using the flag number specified for the current priority group. It is a requirement that the first vehicle loaded be the widest vehicle from the available stock that will fit into the space considered. The search starts from the left forward position.

LOAD (IAC, PAY, LIST, NVT, WT, NPAX, NSIM, NINA)

Subprogram called by FINL to load the aircraft from the available priority stock list specified.

Input arguments:

IAC	aircraft number
PAY	maximum payload for aircraft and onload base from the payload/time-out table
LIST	flag number for stock list in priority group
NVT	number of vehicles

Output arguments:

WT	total weight loaded on aircraft
NPAX	number of personnel on aircraft
NSIM	number of aircraft of the current type that may be loaded with the same configuration. NSIM = 1000000 if aircraft is empty; NSIM = 0 if more than 50 vehicles loaded. The program keeps the list for only 50 vehicles.
NINA	number of cargo vehicles loaded on aircraft

Fig. C.20—Routine LOAD

The space available for stowing a vehicle is defined by a moving left line XF and a moving right line XT, which represent the distances from the left side of the compartment. The difference between the two expresses the maximum width available at the moment an item is being stowed. The length of the space available for stowing an item of equipment is the difference between the length of the compartment and the line SL. SL marks the momentary forward position at which equipment can be stowed and is the distance of that point from the front of the compartment. The height of the vehicle is checked, and its weight is checked against the unused weight capacity of the aircraft. (The vehicles' dimensions include the required clearances.) Initially, $XF = 0$, $XT = \text{width of compartment}$, and $SL = 0$.

When the selection of vehicle is made, the left line XF is recorded in XL, and the right line ($XF + \text{width of vehicle}$) is recorded in XR for that vehicle. The remaining weight and the quantity in the stock list and priority stock list are reduced. In addition, the

distance from the front of the aircraft compartment to the end of the stowed vehicle is stored in the STA array. The subscript for XL, XR, STA is recorded in NTL. The order within NTL is such that $NTL(I) \dots NTL(I+n)$ point to XL values in ascending order. NSAVE records the vehicle number.

Vehicles are stowed side by side until nothing will fit in the row. If there is enough width for passengers, the length of this space is added to the sum of space reserved for passengers up to this point. The rear line of the vehicle preceding the passengers is extended to the right limit of the aircraft.

After a row is completed, the program checks to see whether or not there is a gap (explained below). If there is no gap, search starts for the STA value that is closest to the front of the compartment. This becomes the SL value. For each vehicle that has that same STA value, STA and NTL are set to zero. Then nonzero values are moved up to consecutive positions in NTL.

Figure C.21 illustrates a placement of vehicles that leads to the creation of a gap. After number 4 vehicle is stowed, there is no space against the forward line for a vehicle, but enough space for passengers. The value of XR_4 is changed to equal that of the width of the aircraft and the "passenger space" is assumed to be filled. Since all vehicles are adjacent to one another, no storage space remains available against the forward line; the program, therefore, finds the new most forward line (in the illustration, the line behind vehicles 2 and 3).

A gap exists when one vehicle is no longer adjacent to the last previous vehicle stowed, that is, when the value of the left line XL_i is greater than that of the right line XR_{i-n} of the previous vehicle. In the example of Fig. C.21, vehicles 2 and 3 have the same STA value. Their lines XL_2 , XR_2 , XL_3 , and XR_3 are removed from the list. The NTL array now points to vehicles 1 and 4 only. Since the left line of vehicle 4, XL_4 , does not coincide with the right line of vehicle 1, XR_1 , a gap exists.

The new space available has a left line equal to XR_1 and a right line equal to XL_4 , and $SL = STA_2$. Vehicle 5 is now stowed. The new NTL list points to vehicles 1, 5, and 4 in ascending order for the XL

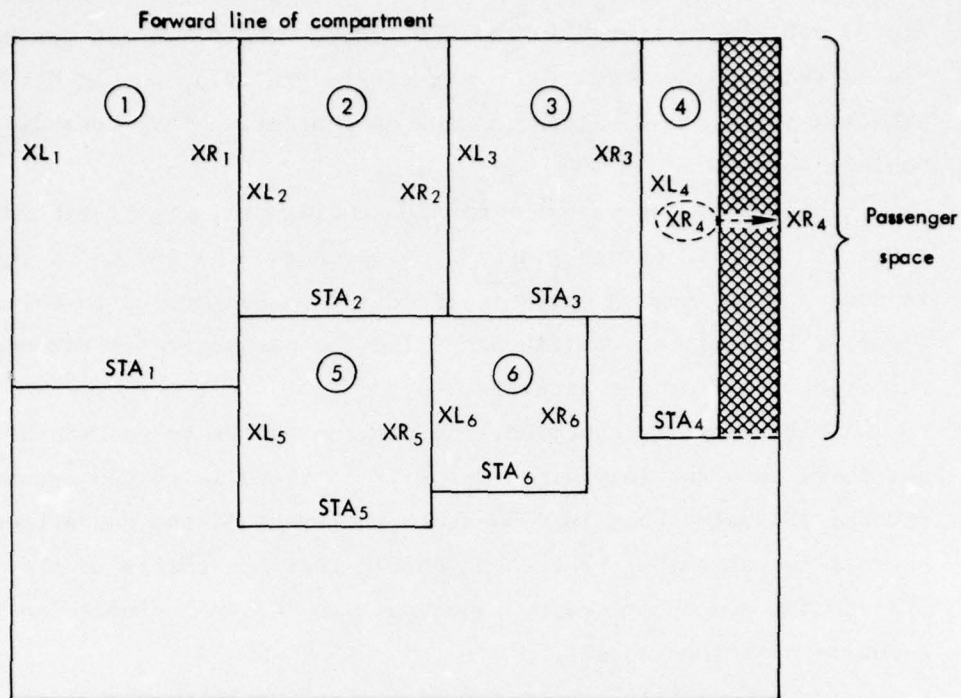


Fig. C.21—Stowing vehicles

lines. There is still a gap between XR_5 and XL_4 . After vehicle 6 is stowed, there are no usable gaps between vehicles 1, 5, 6, and 4. The program extends line STA_4 to the left to XR_6 . Vehicle 4 will have XL_4 coincide with XR_6 when the program gets to consider line STA_4 .

Now again the most forward line is sought, which is behind vehicle 1, and the process continues.

When nothing more can be loaded, the program calculates the number of passengers from the accumulated passenger spaces. Then it checks the current configuration of vehicles and passengers against the remaining stock to determine the number of aircraft of the same type, at the same base, with the same priority flag that may be loaded with the same configuration.

The internal program list can record only 50 vehicles loaded. If there are more, then a value of zero is returned. If the plane is empty, a value of one million is returned.

Routines PDEL and PCDEL (Figs. C.22 and C.23) print information about deliveries.

PDEL (NVT, TWT, NPAXPG, NPGC, POUT)

Subprogram called by MAIN to print deliveries.

Input arguments:

NVT	number of vehicles
NPGC	priority group number

Output arguments:

TWT	cargo weight
NPAXPG	number of personnel
POUT	array used for printing graph of cargo and personnel

Fig. C.22—Routine PDEL

PCDEL (NVT, WTCT, NPAXT, POUT, NACF, IVSP4, IVSP6, NACT)

Subprogram called by MAIN to print cumulative deliveries for one case.

Input arguments:

NVT	number of vehicles
WTCT	total cargo weight
NPAXT	total number of personnel
NACF	number of aircraft personnel
IVSP4	control for printing aircraft release times
IVSP6	control for printing activity at offload base
NACT	number of aircraft types

Output argument:

POUT	array used for printing graph of cargo and personnel
------	--

Fig. C.23—Routine PCDEL

INPUT DATA FORMATS

Overview

Data cards are entered by card type. The beginning of this report has the detailed description of the data. Card formats and program variable names follow.

There must be a blank card between each card type group. If an optional card type has been omitted, a blank card must be inserted. For example, if there are no type 04 cards, the deck would have

Type 02 cards

Blank card

Type 03 cards

Blank card

Blank card

Type 05 cards

.
.
.

Input Card Formats:

1. Title card, card type 00; read by FRST; one card.

Columns	Contents	Variable Name
1-2	00	ISW
3-80	title	(read into format statement)

2. Parameter card, card type 00; read by FRST; one card.

Columns	Contents	Variable Name	Format
1-2	00	ISW	I2
6-10	clearance width	CLW	F5.0
11-15	clearance height	CLH	F5.0
16-20	clearance length	CLL	F5.0
21-25	hours increments, default 12 allows 100 days for printing	HRSINC	F5.0

2. Parameter card continued:

The next 11 fields control optional output

Columns	Contents	Variable Name	Format
26-30	distance table	IVSP1	I5
31-35	vehicle data	IVSP2	I5
36-40	priority group graphs	IVSP3	I5
41-45	aircraft release times	IVSP4	I5
46-50	priority group composition	IVSP5	I5
51-55	activity at offload base	IVSP6	I5
56-60	deliveries for each initial sortie	IVSP7	I5
61-65	deliveries for sortie after all aircraft have made their initial sortie	IVSP8	I5
66-70 }	available for future use	IVSP9	I5
71-75 }		IVSP10	I5
76-80 }		IVSP11	I5

3. Vehicle data, card type 01; read by FRST; one card per vehicle.

Columns	Contents	Variable Name	Format
1-2	01	ISW	I2
3-4	blank	--	2X
5-10	vehicle name	NAMEV (I)	A6
11-15	vehicle number	NV (I)	I5
16-20	width*	VWD (I)	F5.0
21-25	height*	VHT (I)	F5.0
26-30	length*	VLENG (I)	F5.0
31-36	weight quantity in	VWT (I)	F6.0
37-40	unit 1	NOV (I,1)	I4
41-44	unit 2	NOV (I,2)	
45-48	unit 3	.	.
49-52	unit 4	.	.

*Program adds clearance values from parameter card.

3. Vehicle data continued:

Columns	Contents	Variable Name	Format
53-56	unit 5	NOV (I,5)	I4
57-60	unit 6	.	.
61-64	unit 7	.	.
65-68	unit 8	.	.
69-72	unit 9		
73-76	unit 10		
77-80	unit 11	NOV (I,11)	I4

4. Vehicle lists, card type 02; read by FRST; one to seven cards per list number. One to five list numbers, if any.

Columns	Contents	Variable Name	Format
1-2	02	ISW	I2
3-4	blank	--	2X
5	list number	LN	I1
6-10	vehicle number,	NSAVE (1)	I5
11-15	up to 15 per	.	.
.	card maximum	.	.
76-80	of 100 vehicles	NSAVE (15)	I5

Vehicle numbers get transferred to LISV (LN,J)

These external vehicle numbers are changed to internal numbers; that is, the subscript for the vehicle in the vehicle data array NV.

5. Base data, card type 03; read by RDNT; one card per base.

Columns	Contents	Variable Name	Format
1-2	03	ISW	I2
3-4	blank	--	2X
5-10	base name	NAME (I)	A6
11-15	base number	NO (I)	I5
16-20	ground time	GRT (I)	F5.0

5. Base data continued:

Columns	Contents	Variable Name	Format
21-27	latitude degrees*	ZLATD (I)	F7.0
28	blank		IX
29-30	latitude minutes	ZLATM (I)	F2.0
31-37	longitude degrees*	ZLONGD (I)	F7.0
38	blank		IX
39-40	longitude minutes	ZLONGM (I)	F2.0

6. Link data, card type 04; read by RDNT; one card per link, if any.

Columns	Contents	Variable Name	Format
1-2	04	ISW	I2
3-5	blank	--	3X
6-10	base number from	NFR	I5
11-15	base number to	NT0	I5
16-20	distance in nautical miles. Blank if link denied	DISTX	F5.0

7. Offload and onload bases, card type 05; read by NETW; one card per base.

Columns	Contents	Variable Name	Format
1-2	05	ISW	I2
3-5	blank	--	3X
6-10	offload base number	IWAR	I5
11-15	onload base numbers	NON (I)	14I5
16-20			
.			
.			
76-80			

* For negative latitude and/or longitude precede the degrees with a minus sign. For negative latitude and/or longitude with *zero degrees*, use *-360 degrees*.

8. Aircraft type data, card type 06; read by NETW; one card per aircraft type.

Columns	Contents	Variable Name	Format
1-2	06	ISW	I2
3-4	blank	--	2X
5-10	aircraft name	ANAME(IAC)	A6
11-15	aircraft number	ANO(IAC)	F5.0
<u>Compartment Dimensions</u>			
16-20	width (inches)	AWID(IAC)	F5.0
21-25	height (inches)	AHT(IAC)	F5.0
26-30	length (inches)	ALENG(IAC)	F5.0
31-35	passenger capacity	EPAX(IAC)	F5.0
36-40	speed (knots)	SPEED(IAC)	F5.0
<u>Ranges (nautical miles)</u>			
41-45	with maximum payload	APAY(IAC,1)	F5.0
46-50	with reduced payload	APAY(IAC,3)	F5.0
51-55	ferry range	APAY(IAC,5)	F5.0
56-63	maximum payload	APAY(IAC,2)	F8.0
64-71	reduced payload ground time per	APAY(IAC,4)	F8.0
72-78	flying hour	GTPFH	F7.0
79-80	number of bases denied	NBDN	I2

9. Base denial cards, card type 06; read by TMBK; as many cards as specified in aircraft type data, card type 06 columns 79-80.

Columns	Contents	Variable Name	Format
1-2	06	--	} 5X
3-5	blank	--	
6-10	base number	NSAVE(I)	15I5
11-15			
.			
.			
76-80			

10. Stock data, card type 07; read by RDST; one card per request.

Columns	Contents	Variable Name	Format
1-2	07	ISW	I2
3-5	blank		3X
6-10	flag number	NFX	I5
11-15	base number	NBX	I5
16-20	quantity	Q(1)	F5.0
21-25	unit	NU(1)	I5
26-30	list number	NL(1)	I5

11. First priority group data, card type 08; read by RDPG; one card per request.

Columns	Contents	Variable Name	Format
1-2	08	ISW	I2
3-5	blank		3X
6-7	flag numbers	NFSP (1)	9I2
8-9		NFSP (2)	
10-11		NFSP (3)	
12-13		NFSP (4)	
14-15		NFSP (5)	
16-17		NFSP (6)	
18-19		NFSP (7)	
20-21		NFSP (8)	
22-23		NFSP (9)	
24-30	quantity	Q (1)	F7.0
31-35	unit	NU (1)	I5
36-40	list number	NL (1)	I5

The program transfers flag numbers from NFSP to NF for each card; thus the flag numbers from the last card are preserved.

12. Aircraft arrivals, card type 09; read by FSDL; one card per aircraft group.

Columns	Contents	Variable Name	Format
1-2	09	ISW	I2
3-5	blank		3X
6-10	base number	NBX	I5
11-15	flag number of first stock list	NFX1	I5
16-20	flag number of second stock list	NFX2	I5
21-25	aircraft type (ID number)	NA	I5
26-30	quantity in groups	NQTY	I5
31-35	hours before first arrival	DELT	F5.0
36-40	hours between arrivals	TBA	F5.0

13. Remaining priority group data, card type 10; read by RDPG.

The card format for the remaining priority groups is the same as for the first priority group data with 10 in columns 1-2.

14. End of case, card type 11; tested in MAIN program. The only information on this card is 11 in columns 1-2.

COMMON VARIABLES

Definitions for COMMON

STA(50): The length from the forward position of the compartment to the end of the stowed vehicle. Used in LOAD. If more than 50 pieces of cargo are loaded, position 50 stores the values.

NTL(50): Pointer to the subscript for arrays XL, XR, STA used to mark the position of a vehicle in the aircraft as stored by LOAD. The pointers are for ascending order in XL.

XL(50): Distance from the left side of the compartment to the left side of a vehicle stored by LOAD.

XR(50): Distance from the left side of the compartment to the right side of a vehicle stored by LOAD. For any vehicle $XR(I) = XL(I) + VWD(I)$.

DIST(50, 50): Distance between two bases, first computed as the great circle distance in RDNT, using longitudes and latitudes entered in card type 03 card. If the type 04 link data card has a distance between links in columns 16-20, it replaces the value in the DIST table. If columns 16-20 are blank, the link value becomes 9,999,999. Subprogram NWLK then computes the shortest distance between all bases. The DIST matrix stores values in its upper triangle only, that is for DIST(I, J) where $J > I$. For links where $J < I$, computation uses DIST(J, I). The DIST matrix has been initialized to zero before the calculation of distances.

NSUBN(50, 50): For each base it is the list of bases in the route from onload to offload base for maximum payload for an aircraft. Set in SUBN from array NP computed in MINT.

FILL(250): Never referenced. It is a filler to accommodate the space needed by variable NSTK(16, 200) which is equivalent to DIST(2002).

AHT(10): Compartment height--in columns 21-25 of card type 06. Read by NETW.

ALENG(10): Compartment length--in columns 26-30 of card type 06. Read by NETW.

ANAME(10): Aircraft name--in columns 5-10 of card type 06. Read by NETW.

ANO(10): Aircraft number--in columns 11-15 of card type 06. Read by NETW.

APAY(10, 5): Aircraft ranges and payload. Data in card type 06, read by NETW. The second dimension stores the following for each aircraft:

1. Range with maximum payload, in columns 41-45.
2. Maximum payload, in columns 56-63.
3. Range with reduced payload, in columns 46-50.
4. Reduced payload, in columns 64-70.
5. Ferry range, in columns 51-55.

AWID(10): Compartment width--in columns 16-20 of card type 06. Read by NETW.

EPAX(10): Aircraft passenger capacity--in columns 31-38 of card type 06. Read by NETW.

GRT(50): Base ground time--in columns 16-20 of card type 03. Read by NETW.

JNOW(10): Pointer to second subscript in NPAC(I, J) table for each aircraft. Set to 1 in FINP. Used and modified in ADEL.

LISV(5, 100): Vehicle list of card type 02. Read by FRST.

NAME(50): Base name in columns 5-10 of card type 03. Read by RDNT. (Variable defined as REAL*8.)

NAMEP(50): Base names in maximum flow route. Printed by SBN2. (Variable defined as REAL*8.)

NAMEV(200): Vehicle name in columns 5-10 of card type 01. Read by FRST. (Variable defined as REAL*8.)

NAOTB(10, 50): Pointer into the array WTMV for the start of information for an aircraft-and-base combination. Value set in NETW upon return from SBN2.

NBASE(50): Internal number for the base associated with a stock list and a flag number. It is a pointer to the array NO, which contains the base numbers. Data come from columns 11-15 of card type 07, read by RDST.

NF(9): Flag numbers associated with a priority group, from two column fields in 6-23 in card type 08 or 10. Read by RDPG.

NFLG(50): Flag number for a stock list, from columns 6-10 in card type 07. Read by RDST.

NL(7): The vehicle list number to use in assembling stock. It references a list number entered by card type 02. NL comes from card columns 29-30 in card type 07. Read by RDST.

If $NL > 0$, the vehicles specified in the vehicle list number are used.

If $NL < 0$, the vehicles specified in the vehicle list number are excluded.

If $NL = 0$, all vehicles are used that are required by the unit in card type 07.

NO(50): Base number in columns 11-15 in card type 03. Read by RDNT.

NODE(50): In TMBK it is set to the list of acceptable bases for an aircraft's route. SBN2 stores bases when computing the route out.

NON(50): List of onload bases entered in card type 05, read by NETW.

NOV(200, 11): List of the required quantity of a vehicle in a unit. Data entered in 4 digit fields in columns 37-80 in card type 01, read by FRST.

NP(50): Route determined for minimum time by MINT.

NPAC(10, 50): For each aircraft, the pointers into the stock lists that may be used to load the aircraft after the first deliveries.

NSAVE(50): Used to store information in different subprograms.

NTL1(50): Used as temporary storage in LOAD.

NU(7): Unit number for priority group in columns 31-35 in card type 08 or 10, read by RDPG.

NV(200): Vehicle number in columns 11-15 in card type 01, read by FRST.

Q(7): The quantity that is the multiple of the stock required by the unit NU in the priority group. The value is in columns 24-30 in card type 08 or 10, read by RDPG.

SPEED(10): Speed of the aircraft in knots in columns 36-40 in card type 06, read by NETW.

TB(50): Minimum time back to base from the offload base. SUBN stores values from array TMIN computed by TMBK. In ADEL, the time back for the current base is recomputed using data from the maximum payload table WTMV.

VHT(200): Vehicle height plus clearance height. Vehicle height is in columns 21-25 in card type 01. Clearance height is in 11-15 in card type 00. Both cards read by FRST, which does the addition.

VLENG(200): Vehicle length plus clearance length. Vehicle length is in columns 26-30 in card type 01. Clearance length is in 16-20 in card type 00. Both cards read by FRST, which does the addition.

VWD(200): Vehicle width plus clearance width. Vehicle width is in columns 16-20 in card type 01. Clearance width is in 6-10 in card type 00. Both cards read by FRST, which does the addition.

VWT(200): Vehicle weight in columns 31-36 in card type 01, read by FRST.

WDEL(200): Cargo weight delivered in the current priority group. The subscript is the time increment for printing deliveries. Values are modified in FSDL and ADEL. The array is cleared in LOAD and PDEL after printing.

WTM(50): Array contains pairs of values for payload and for the time to fly a load. WTM computes the values for one aircraft at distances ranging from ferry range to range for maximum payload. List terminated with -1, if there are fewer than 50 entries.

WTMV(2000): Payload and time out table for an aircraft and onload base combination. The first entry is the maximum flow, computed from maximum payload divided by the sum of time out and time back. Following this are pairs of values where the first is the time to deliver the load. The time includes ground time at the bases enroute to the offload base and the required aircraft ground time. The second is the load. These pairs are in descending order of payload. The last entry for an aircraft/base combination is negative. The array NAOTB contains the subscript for the starting point for the aircraft/base combination. NETW stores the array from WTM.

GTPFH: Ground time per flying hour for aircraft in columns 72-78 in card type 06, read by NETW.

MDEL(200): Personnel delivered in the current priority group. This array is analogous to WDEL for weight delivered.

WDELC(200): Cumulative cargo weight delivered for all priority groups. The subscript is the time increment for printing deliveries.

MDELC(200): Cumulative personnel delivered for all priority groups. This array is analogous to WDELC for cargo.

HRSINC: Hour increment for recording the simulation results for output. The value is in columns 21-25 in card type 00, read by FRST. If the field is blank, the default value is 12. The default allows printing for 100 days of simulation. If more days are needed use a larger increment.

NACARR(200): Number of aircraft that have arrived in a time increment at the offload base.

NACDEP(200): Number of aircraft available for departure from the offload base.

NORMAN(10): Number of sorties for an aircraft.

Definitions for Variables Equivalent to COMMON

FLOW(50): Shares storage with all of STA: Maximum flow computed in SUBN.

NROUT(50): Shares storage with all of NTL. Route back from the offload base to an onload base. Used temporarily in SBN2.

CL(50): Shares storage with all of XL. Set to zero in NWLK. Set to zero and used in MINT to compute the distance in the route for the minimum time network. This distance is used in SUBN for computing the maximum payload.

TMIN(50): Shares storage with all of XR. Used for computation of minimum time routes in MINT.

ARR(1000): Shares storage with DIST(1,1) to DIST(50,20). This array has the time the aircraft is available for redeployment. The array is in ascending sequence for time. The time includes the time back to the onload base.

NTARR(1000): Shares storage with DIST(1,21) to DIST(50,40). This is the aircraft type that is available for redeployment. This array has a one-to-one correspondence with ARR.

NSTK(16,200): Shares storage with DIST(2,41) to DIST(50,50); NSUBN(1,1) to NSUBN(50,50); and FILL(1) to FILL(201). Each row of this matrix is a stock list indicating the available quantity of each vehicle type. Subprogram STLS computes the quantities from the data entered in card types 01, 02, and 07. Array NFLG has the flag number for the row and NBASE has the base that has the stock. Row 16 is used to aggregate quantities for the current priority group.

Definitions for COMMON/LAB01/

IVSP7: Control for the optional output for the initial sortie of each aircraft. Enter a one in column 60 of the card type 00 to print the onload base, the load, and the personnel carried by the aircraft on its first flight to the offload base. Referenced in FRST and FSDL.

IVSP8: Control for the optional output for all sorties of each aircraft after its initial sortie. Enter a one in column 65 of the card type 00 to print the onload base, the load, and the personnel carried to the offload base. Referenced in FRST and ADEL.

IVSP9 } Available for additional parameters in card type 00. Use in
IVSP10 } columns 66-70, 71-75, 76-80. Referenced in FRST.
IVSP11 }

References to COMMON Variables:

Variables	Routines						
STA	LOAD*						
NLT	LOAD*						
XL	LOAD*						
XR	LOAD*						
DIST	MINT	NWLK*	RDNT*				
NSUBN	SBN2	SUBN*					
FILL							
AHT	LOAD	NETW*					
ALENG	LOAD	NETW*					
ANAME	MAIN	FSDL	NETW*	PCDL			
ANO	FSDL	NETW*					
APAY	NETW*	PAYL	SUBN	WTIM			
AWID	LOAD	NETW*					
EPAX	LOAD	NETW*					
GRT	FSDL	ADEL	MINT	RDNT*			
JNOW	FINP*	ADEL*					
LISV	FRST*	STLS					
NAME	FSDL	NWLK	RDNT*	RDST	SBN2		
NAMEP	SBN2*						
NAMEV	FRST*	PRPG					
NAOTB	FINP	ADEL	FINL	NETW*			
NBASE	FINP	FSDL	ADEL	RDST*			
NF	FINP	RDPG*					
NFLG	FINP	FSDL	RDST*				
NL	RDPG*	RDST*	STLS				
NO	FSDL	NETW	RDNT*	RDST	TMBK		
NODE	MINT	SBN2*	TMBK*				
NON	NETW*	SUBN					
NOV	FRST*	STLS					
NP	MINT*	RDST	SUBN				
NPAC	FINP*	ADEL					
NSAVE	FRST*	FINL	LOAD*	SBN2	SUBN*	TMB*	
NTL1	LOAD*						
NU	RDPG*	RDST*	STLS				
NV	FRST*	PRPG					
Q	RDPG*	RDST*	STLS				
SPEED	NETW*	SUBN	WTIM				
TB	ADEL*	NETW	SBN2	SUBN*			
VHT	FRST*	LOAD	PRPG				
VLENG	FRST*	LOAD	PRPG				
VWD	FRST*	LOAD	PRPG				
VWT	FRST*	FSDL	ADEL	LOAD	PRPG		
WDEL	FSDL*	ADEL*	PDEL*				
WTM	NETW	WTIM*					
WTMV	FINP	ADEL	FINL	NETW*			
GTPFH	MINT	NETW*					
MDEL	FSDL*	ADEL*	PDEL*				
WDEL	FSDL*	ADEL*	PCDEL				
MDEL	FSDL*	ADEL*	PCDEL				
HRSINC	FRST*	FSDL*	ADEL	PCDEL	PDEL		
NACARR	FSDL*	ADEL*	PCDEL				
NACDEP	FSDL*	ADEL*	PCDEL				
NORMAN	MAIN	FSDL*	ADEL*				
FLOW = STA(1)		SUBN*					
NROUT = NTL(1)		SBN2*					
CL = XL(1)		MINT*	NWLK*	SUBN			
TMIN = XR(1)		MINT*	NWLK*	SUBN			
ARR = DIST(1)		FSDL*	ADEL*	PCDEL			
NTARR = DIST(1001)		FSDL*	ADEL*	PCDEL			
NSTK = DIST(2002)		FINL*	LOAD*	PRPG	RDPG	RDST*	STLS*

* Value stored into variable.

PROGRAM MODIFICATIONS

New Options

Subject to the user's control, a new printout is available for the sorties of each aircraft. There is one line of output giving a number, the aircraft, the onload base, total weight aboard, unused weights, cargo weight, and the number of personnel carried.

For initial sorties the above mentioned number is a sequential number within the group of aircraft arriving. For subsequent sorties it is the sortie number.

The initial sorties will be printed out if there is a one in column 60 in the card type 00 parameter card.

The subsequent sorties will be printed out if there is a one in column 65 in the card type 00 parameter card.

This extends the number of optional tables from 6 to 8. Eventually the program could have 11 options by using columns 70, 75, and 80. These are read, but not listed.

Subprogram FSDL computes and prints the initial sorties and ADEL does the same for subsequent sorties.

A labeled common block has been added to ADEL, FRST, and FSDL to store the print control switches.

COMMON/LAB01/IVSP7, IVSP8, IVSP9, IVSP10, IVSP11

Program Changes to Correct Problems

Wherever changes have been made, there will be a comment card in the source code with an explanation and a date.

ADEL

REAL*8 AZ,NBZ has been added, since the names of the aircraft and the bases have six characters.

FRST

The read statement and format 111 have been changed to input six more options on the card type 00 parameter card. (This is an addition rather than a correction.)

FSDL

1. REAL*8 AZ,NBZ,ERR has been added since the names of the aircraft, base, and error may have six characters. Currently ERR will have only four characters because it is set by ATHRUZ.
2. Format 101, which was not referenced and was the same as 100, has been removed.
3. Formats 108 and 109 have been changed to double space.
4. For card type 09, if the aircraft identification is incorrect, check the base before printing an error message.
5. Start a new page for each aircraft.

NETW

1. Added format 112 to print message if onload base number in card type 05 is not correct. Program continues as before.
2. In reading card type 05, the program did not check for a limit of 50. If there were more than 42 bases, it would read 56 and destroy the beginning of the NØV array.
3. If the number of onload bases is a multiple of 14, the list will not terminate properly. The program relied on a blank base to signal the end.

Corrections have been made between statements 258 and 260.

PCDEL

The time increment computed is checked by the program to determine whether it is greater than the maximum dimension of 200. If yes, it is set to 200; a message is printed when the table is finished. Other subprograms make the test. This omission in PCDEL caused the program to end abnormally.

RDNT

There has been a change made in format 101, which reads card type 03, to conform to the description of the input. Two checks were inserted for the values of latitude and longitude. If the degrees are negative, the minutes are set negative. Since the IBM 360 and 370 computers store zero as a positive number, a special convention of -360 has to be used for the negative latitude or longitude where the degrees are zero. For example, -0 degrees 40 minutes would be entered as -360 degrees 40 minutes. After the test for negative degrees, the program tests for degrees of 360 and resets to zero.

TMBK

Format 101, which reads the bases denied to an aircraft in card type 06, has been changed. The format was incorrect if more than two bases were denied.

Problems Not Resolved

There may be errors not detected if the user fails to insert a blank card between card types.

In some cases the program stops. In others, if the card type is greater than the previous one, the program continues. This means that the first card of the new type has been read with the wrong format and the data will not be processed.

The following subprograms read data:

FRST	card types	00, 01, 02
RDNT	card types	03, 04
NETW	card types	05, 06
TMBK	card type	06 (bases denied)
RDST	card type	07
RDPG	card type	08 or 10
FSDL	card type	09

In subprogram FINL after statement 300, there is a DO 400 I=1, 30 loop. The assumption is that the maximum flow table will not have more than 30 entries for a base/aircraft combination. If it has more than 30, the duration of the flight will be--erroneously--that of the previous sortie.

In the same loop, there should be a check that the subscript NXX be less than 2000, the dimension of WTMV.

Program FSDL should (but does not) do further checking on the flag numbers. It does not verify that the flag numbers are in the array NF for the current priority group. This could be done after statement 485.

Suggestions for Possible Program Changes

There are two areas in the loading process where computing time might be reduced. The first is in the selection of the cargo vehicle to load. The algorithm chooses the widest vehicle.

- a. Arrange the cargo vehicle lists in descending order of width. The user could do this for all type 01 cards after the first one, which must be personnel. It would save program time to rearrange lists.
- b. Modify subprogram FRST, which reads the vehicle data. The program would check the order and arrange it in proper sequence.
- c. Modify subprogram LOAD in the DO 1250 loop. As soon as a vehicle meets all of the criteria, it would be the best one. It will be as wide or wider than any vehicle following it in the list.

Insert GO TO 396 ahead of the 1250 CONTINUE statement.

The tests in statements 1242 to 1245 should be done in a different sequence. Check width VWD first, then length VLENG and height VHT, with weight VWT last. The dimension of the vehicle may be more of a bind than its weight when the program is trying to fit it into an area.

- d. In subprogram ADEL, after each sortie, the new redeployment time has to be inserted into the array ARR to maintain the ascending order on time. Statements 690 to 700 compare values. A binary search may be faster.
- e. Subprogram FSDL orders ARR, too. However, it does it only once after all planes have arrived with their initial deliveries, so this may not prove unduly inefficient.
- f. There are places in the code where an assignment in a DO loop could be done outside of it. The optimizing computer corrects this.
- g. The author of the original program appears to have adapted it from a still earlier one. Some local variables are not referenced, or are assigned a value which serves no purpose. An example of the latter is WINDF which is an argument in subprogram MINT.
- h. Subprograms PDEL and PCDEL.
In the print of the graph for deliveries of personnel and cargo, put a heading above the graph to the effect that an asterisk denotes materiel and a dot denotes personnel.
- i. Subprogram PCDEL--aircraft availability printout.
Instead of printing DAY HOUR and repeating values--print the DAY HOUR and a count of the number of aircraft for that period. It would be easier to read.
- j. ATHRUZ(A,B).

This subprogram has been written in assembly language to place the alphanumeric character specified in the second argument in the variable name in the first argument. All of the calls to the program use six characters for the alphanumeric, but the variable is single precision, which means it can hold only four characters.

The calling program then uses an assignment statement to transfer the variable. This subprogram was necessary when the original program was written. It can be replaced by declaring a labeled COMMON with a list of variables that hold the desired alphanumeric characters. Use BLOCK DATA to initialize the values.

Delete each call and change the assignment statement that follows. Some variables should be REAL*8.

ATHRUZ called by FSDL, PCDEL, and PDEL. FSDL uses 'ERROR' which could be a local variable. PCDEL and PDEL use blanks, numeral ones, dots, and asterisks.

Appendix D

PROGRAM LISTING

INTRODUCTION

This appendix contains listings of the main program and the sub-routines that constitute the Army Deployment Simulator. The main program and subroutines, compiled and executed on an IBM 370/158 computer, are written primarily in FORTRAN IV. The single exception is the sub-routine ATHRUZ, which is written in OS assembly language. The sub-routines appear in alphabetical order following the main routine, except for ATHRUZ, which is last.

See Appendix C for the section on "Program Modifications." Note that there are problems that have not been resolved.

```

      MAIN PROGRAM FOR DEPLOYMENT SIMULATOR
      DIMENSION AHT(10),ALENG(10),ANAME(10),ANC(10),APAY(10,5),ARR(1000)
      1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLCH(50),GRE(50),JNOW(10),
      2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(50),NACIS(10,50),NBASE(50),
      3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NOV(20,11),NP(50),
      4,NPAC(10,50),NRUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
      5,NTABR(1000),NIL(50),NTLI(50),NU(7),NV(200),O(7),SPEED(10),STA(50)
      6,TB(50),TWIN(50),VHT(200),VLENG(200),VMD(200),VWT(200),WDEL(200),
      7,WPM(50),WTMV(2000),XL(50),XR(50)
      8,WDEL(200),WDELCL(200),WDELCLC(200)
      DIMENSION POUT(51)
      DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
      DIMENSION FILL(250)
      REAL*8 ANAME,NAMEV,NAMEP
      COMMON STA,NTL,XL,XR,DIST,NSUBN
      COMMON FILL
      COMMON AHT , ALENG , ANAME , ANC , APAY , AWID ,
      COMMON EPAX , GRE , JNOW , LISV , NAME , NAMEP ,
      COMMON NAMEV , NACIS , NBASE , NF , NPLG , NL ,
      COMMON NO , NODE , NOV , NP , NPAC ,
      COMMON NSAVE , NTLI , NU , NV , O ,
      COMMON SPEED , TB , VHT , VLENG , VMD , VWT ,
      COMMON WDEL , WTM , WTMV , GTPPH , MDEL ,
      COMMON WDELCL , WDELCLC , HESINC ,
      COMMON NACARR , NACDEP , NORMAN
      EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRUT(1)),(XL(1),CL(1)),
      1 (KR(1),TWIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTABR(1)),
      2 (DIST(2002),NSTK(1))
      DIMENSION ISTEVE(14342)
      EQUIVALENCE (ISTEVE(1),STA(1))
      DO 500 I=1,14342
      ISTEVE(I)=0
      500 CONTINUE
      101 FORMAT (1H 10F12.1 )
      102 FORMAT (1H 25I5 )
      103 FORMAT (16H1 TOTAL SORTIES )
      104 FORMAT (34H0 AIRCRAFT NUMBER OF SORTIES )
      105 FORMAT (1H0 2X, A6,I10)
      106 FORMAT (I2)
      CALL ZERSET(209,256,-1,1)
      CALL PRST(NVT,IVSP1,IVSP2,IVSP3,IVSP4,IVSP5,IVSP6)
      CALL NETW(IWAB,NACT,NNODES,IVSP1)
      150 MPGC = 1
      MPACT = 0
      WTCT = 0.0
      CALL EDST(NSTL,NVT,NNODES)
      CALL RDPG (NVT,ISW,NPGC)
      CALL PRPG (NVT,THT,NPAKPG,IVSP5)
      MPACT = NPAIT + NPAKPG
      WTCT = WTCT + NVT
      CALL PSDL (NVT,NNODES,NACT,NSTL,NACP,IKAB)
      CALL FIMP (NACT,NSTL)
      CALL ADEL (NVT,NACP,IWAB)
      IP (IVSP3) 200,200,180
      180 CALL PDEL(NVT,THT,NPAKPG,NPGC,POUT)

```



```

200 NP3C = NP3C + 1
    CALL RDPG (NVT, ISW, MPGC)
    IF (ISW - 1) 210, 300, 210
210 CALL PRPG (NVT, IWT, NPXPG, IVSP5)
    NPXIT = NPXIT + NPXPG
    WTCT = WTCT + IWT
    CALL FINP (NACT, NSTL)
    CALL ADEL (NVT, NACP, IWAR)
    IF (IVSP3) 200, 200, 220
220 CALL PDEL (NVT, IWT, NPXPG, NE3C, POUT)
    GO TO 200
300 CALL PCDEL (NVT, WTCT, NPXIT, POUT, NACP, IVSP4, IVSP6, NACT)
    WRITE (6, 103)
    WRITE (6, 104)
    DO 350 K = 1, NACT
350 WRITE (6, 105) ANAME(K), NORMAN(K)
    READ (5, 106, END=12345)
    GO TO 150
12345 CALL EXIT
      STOP
      END

```

```

MAIN0570
MAIN0580
MAIN0590
MAIN0600
MAIN0610
MAIN0620
MAIN0630
MAIN0640
MAIN0650
MAIN0660
MAIN0670
MAIN0680
MAIN0690
MAIN0700
MAIN0710
MAIN0720
MAIN0730
MAIN0740
MAIN0750
MAIN0760
MAIN0770

```

```

CADEL      SUBROUTINE TO LOAD AIRCRAFT AND RECORD DELIVERIES
      SUBROUTINE ADEL (NVT,NACF,IWAF)
      DIMENSION ADEL(10),ALENG(10),ANAME(10),ANCI(10),APAY(10,5),ARR(1000)
      1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLOW(50),GRT(50),JNCH(10),
      2,LISV(5,100),NAME(50),NAHEP(50),NAHEV(200),NAOTB(10,50),NBASE(50),
      3,NP(9),NPLG(50),NL(7),NO(50),MODE(50),NON(50),NOV(200,11),NP(50),
      4,NPAC(10,50),NRCT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
      5,NFARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
      6,FB(50),TRIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
      7,WPM(50),WTMV(2000),XL(50),XE(50)
      8,WDEL(200),WDELCL(200),WDELCLC(200)
      DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
      DIMENSION FILL(250)
      REAL*8 ANAME,NAME,NAHEV
      REAL*8 NAMEP
      COMMON STA,NTL,XL,XR,DIST,NSUBN
      COMMON FILL
      COMMON ANT , ALEN3 , ANAME , ANO , APAY , AWID
      COMMON EPAI , GRT , JNCH , LISV , NAME , NAMEP
      COMMON NAHEV , NAOTB , NBASE , NF , NPLG , NL
      COMMON NO , NDE , NON , NOV , NP , NPAC
      COMMON NSAVE , NTL1,NU,NV,Q
      COMMON SPEED , TB , VHT , VLENG , VWD , VWT
      COMMON WDEL , WTM , WTMV , GTPFH , WDEL
      COMMON WDELCL , WDELCLC , HRSINC
      COMMON NACARR,NACDEP,NORMAN
      EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRCT(1)),(XL(1),CL(1)),
      1 (KR(1),TRIN(1)), (DIST(1),ARR(1)), (DIST(1001),NTARR(1)),
      2 (DIST(2002),NSTK(1))
      C--> ADD LABELED COMMON TO HOLD 5 MORE PRINT SWITCHES 6/12/75
      COMMON /LAB01/ IVSP7,IVSP8,IVSP9,IVSP10,IVSP11
      C-->
      C--> ADDED THE FOLLOWING 6/12/75
      REAL*8 AZ,NEZ
      C
      C
      C--> 110 & 111 FORMATS ADDED 6/12/75
      110 FORMAT (/ / 1H ,29X,9H TOTAL /
      1 11X, 6HAIR ,3X, 6HONLOAD,3X, 9H WEI:HT,3X, 9H UNUSED ,3X,
      2 9H CARGO ,3X, 9HNUMBER OP /
      3 11X, 6HCRAFT ,3X, 6HBASE ,3X, 9H LOADED,3X, 9HCAPACITY ,3X,
      2 9H WEIGHT,3X, 9HPERSONNEL / )
      111 FORMAT (1H ,17,3X,A6,3X,A6,3F12.1,I12)
      NSW = 1
      NLDG = 1
      C
      C--> SET FOR PRINTING DELIVERIES IF IVSP7 NOT ZERO 6/12/75
      C--> INITIALIZE THE NUMBER OF SORTIES
      IF (IVSP8.EQ.0) ASSIGN 5450 TO KSWP
      IF (IVSP8.NE.0) ASSIGN 5410 TO KSWP
      ASSIGN 5420 TO KSWH
      NSORTY = 0
      DO 50 K=1,10
      NSORTY = NSORTY + NORMAN(K)
      50 CONTINUE

```

```

120 IAC = NTAB(NLDG)
130 JX = JNOW(IAC)
    IF (JI - 50) 150,150,200
150 LIST = NPAC(IAC,JX)
    IF (LIST) 200,200,250
200 NLDG = NLDG + 1
    IF (NLDG - NACP) 120,120,225
225 RETURN
250 N08 = NBASE(LIST)
C
C-->      STORE NAME OF A/C AND BASE      6/12/75
        AZ = ANAME(IAC)
        NBZ = NAME(N08)
        WTHX = APAY(IAC,2)
        CALL PINL(NSW,IAC,LIST,N08,NVT,WTC,NPAX,TTG)
        ZNPAX = NPAX
        WTC = WTC - ZNPAX * VWT(1)
        NSW = 0
        IF (WTC) 300,300,400
300 JNDW(IAC) = JNDW(IAC) + 1
    GO TO 130
400 NPL = NACTB(IAC,N08)
C
C-->      CHECK FOR PRINTING DELIVERIES & HEADINGS      6/12/75
        30 TO KSWP, (5410,5450)
5410 GO TO KSWH, (5420,5430)
C
C-->      PRINT HEADINGS
5420 ASSIGN 5430 TO KSWH
        WRITE (6,110)
5430 WDELTA = WTHX - WTC
        WSCRTY = WSCRTY + 1
        WRITE (6,111) WSCRTY,AZ,NBZ,WTC,WDELTA,WTC,NPAX
C
C-->      JUMP HERE TO BYPASS PRINTING      6/12/75
5450 CONTINUE
    IF (WTHV(NPL+2) ) 405,410,410
405 TB(1) = - WTHV(NPL+2)/WTHV(NPL) - WTMV(NPL+1)
    30 TO 420
410 TB(1) = WTHV(NPL+2) / WTHV(NPL) - WTHV(NPL+1)
420 TX = TB(1) + TTC + ABR(NLDG)
    NDT = (TX / HRSINC) + 1.0
    IF (NDT - 200) 560,560,550
550 NDT = 200
560 WDEL(NDT) = WDEL(NDT) + WTC
        FORHAN(IAC) = FORHAN(IAC) + 1
        WDEL(NDT) = WDEL(NDT) + WTC
        WDEL(NDT) = WDEL(NDT) + NPAX
        WDEL(NDT) = WDEL(NDT) + NPAX
        NACDEP(NDT) = NACDEP(NDT) + 1
        TXB = TX - GRT (IWAR)
        NDTB = (TXB / HRSINC) + 1.
        IF (NDTB - 200) 565,565,564
564 NDTB = 200
565 NACARR(NDTB) = NACARR(NDTB) + 1
        NLOP = NLDG + 1
        IF (NLOP - NACP) 690,670,680

```

ADEL0570
 ADEL0580
 ADEL0590
 ADEL0600
 ADEL0610
 ADEL0620
 ADEL0630
 ADEL0640
 ADEL0650
 ADEL0660
 ADEL0670
 ADEL0680
 ADEL0690
 ADEL0700
 ADEL0710
 ADEL0720
 ADEL0730
 ADEL0740
 ADEL0750
 ADEL0760
 ADEL0770
 ADEL0780
 ADEL0790
 ADEL0800
 ADEL0810
 ADEL0820
 ADEL0830
 ADEL0840
 ADEL0850
 ADEL0860
 ADEL0870
 ADEL0880
 ADEL0890
 ADEL0900
 ADEL0910
 ADEL0920
 ADEL0930
 ADEL0940
 ADEL0950
 ADEL0960
 ADEL0970
 ADEL0980
 ADEL0990
 ADEL1000
 ADEL1010
 ADEL1020
 ADEL1030
 ADEL1040
 ADEL1050
 ADEL1060
 ADEL1070
 ADEL1080
 ADEL1090
 ADEL1100
 ADEL1110
 ADEL1120
 ADEL1130


```

670 IF (TX - ARR(NLDP) ) 671,671,672
671 ARR(NLDG) = TX
    GO TO 900
672 NTARR(NLDG) = ARR(NLDP)
    NTARR(NLD:) = NTARR(NLDP)
    ARR(NLDP) = TX
    NTARR(NLDP) = IAC
    GO TO 900
680 ARR(NLDG) = TX
    GO TO 900
690 DO 700 I = NLDG, NACF
    IF (ARR(I) - TX) 700,700,750
700 CONTINUE
    IX = NACF - 1
    GO TO 770
750 IF (I - NLDP) 671,671,752
752 IX = I - 2
770 DO 800 K = NLDG, IX
    ARR(K) = ARR(K+1)
    NTARR(K) = NTARR(K+1)
    ARR(IX + 1) = IX
    NTARR(IX + 1) = IAC
    GO TO 120
900 END

```

```

ADEL1140
ADEL1150
ADEL1160
ADEL1170
ADEL1180
ADEL1190
ADEL1200
ADEL1210
ADEL1220
ADEL1230
ADEL1240
ADEL1250
ADEL1260
ADEL1270
ADEL1280
ADEL1290
ADEL1300
ADEL1310
ADEL1320
ADEL1330
ADEL1340
ADEL1350
ADEL1360
ADEL1370

```

```

C FINL SUBROUTINE TO FIND LOAD AND TIME
SUBROUTINE FINL(NSW,IAC,LIST,NOB,NVT,WTC,NPAX,TT0)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10),ARR(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLCW(50),GRT(50),JNOW(10),
  2 LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAOTB(10,50),NBASE(50),
  3 NP(9),NFLG(50),NL(7),NO(50),NODE(50),NOV(200,11),NP(50),
  4 NPAC(10,50),NRQUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5 NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
  6 TAB(50),TMIN(50),VHT(200),VLEN3(200),VWD(200),VWT(200),WDEL(200),
  7 WTH(50),WTHV(2000),XL(50),XR(50)
  8 ,WDEL(200),WDELCL(200),WDELCLC(200)
  DIMENSION MACARR(200),MACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AWID
  COMMON EPAX , GRT , JNOW , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTB , NBASE , NF , NFLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE,NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLEN3 , VWD , VWT
  COMMON WDEL , WTM , WTHV , GTPPH , MDEL
  COMMON WDELCL , WDELCLC , HRSINC
  COMMON MACARR,MACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRQUT(1)),(XL(1),CL(1)),
  1 (XR(1),TMIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  IF (NSW) 104,104,199
  104 IF (IAC - IACL) 199,105,199
  105 IF (LIST - LISL) 199,106,199
  106 IF (NSIM) 199,199,107
  107 IF (NINA) 115,115,108
  108 DO 11C K = 1, NINA
    NSP = NSAVE(K)
    NSTK(LIST,NSP) = NSTK(LIST,NSP) - 1
    110 NSTK(16,NSP) = NSTK(16,NSP) - 1
    115 NSTK(LIST,1) = NSTK(LIST,1) - NPAX
    NSTK(16,1) = NSTK(16,1) - NPAX
    NSIM = NSIM - 1
    GO TO 500
  199 NPL = NAOTB(IAC, NOB)
    PAY = WTHV(NPL + 2)
    IF (PAY) 200,300,300
  200 PAY = - PAY
  300 CALL LOAD(IAC,PAY,LIST,NVT,WTC,NPAK,NSIM,NINA)
    DO 400 I = 1, 30
      NIX = NPL + 2 + I
      WTB = WTHV(NIX)
      IF (WTC - WTB) 400,390,385
      385 IF (WTB) 387,387,386
      386 TTC = WTHV(NIX - 3)
      GO TO 450
      387 IF (WTC + WTB) 390,390,386
      390 TTC = WTHV (NIX - 1)

```

FINL0010
 FINL0020
 FINL0030
 FINL0040
 FINL0050
 FINL0060
 FINL0070
 FINL0080
 FINL0090
 FINL0100
 FINL0110
 FINL0120
 FINL0130
 FINL0140
 FINL0150
 FINL0160
 FINL0170
 FINL0180
 FINL0190
 FINL0200
 FINL0210
 FINL0220
 FINL0230
 FINL0240
 FINL0250
 FINL0260
 FINL0270
 FINL0280
 FINL0290
 FINL0300
 FINL0310
 FINL0320
 FINL0330
 FINL0340
 FINL0350
 FINL0360
 FINL0370
 FINL0380
 FINL0390
 FINL0400
 FINL0410
 FINL0420
 FINL0430
 FINL0440
 FINL0450
 FINL0460
 FINL0470
 FINL0480
 FINL0490
 FINL0500
 FINL0510
 FINL0520
 FINL0530
 FINL0540
 FINL0550
 FINL0560

FINL0570
FINL0580
FINL0590
FINL0600
FINL0610
FINL0620

30 TO 450
400 CONTINUE
450 IACL = IAC
LISTL = LIST
500 RETURN
END


```

CFIMP SUBROUTINE TO FILL IN NPAC TABLE
SUBROUTINE FIMP(NACT,NSTL)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10,5),ARR(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLCW(50),JRT(50),JNOW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(50),NAOTB(10,50),NBASE(50),
  3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NCON(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBM(50,50),
  5,NTHRR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPRED(10),STA(50),
  6,TB(50),THIN(50),VHT(200),VLEN(200),VMT(200),VWD(200),WDEL(200),
  7,WTH(50),WTHV(200),XL(50),XR(50)
  8,WDEL(200),WDELCL(200),WDELCLC(200)
  DIMENSION MACARR(200),MACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  COMMON STA,NTL,XL,XR,DIST,NSUBM
  COMMON FILL
  COMMON AHT, ALENG, ANAME, ANO, APAY, AWID
  COMMON EPAX, GET, JNOW, LISV, NAME, NAMEP
  COMMON NAMEV, NAOTB, NBASE, NF, NPLG, NL
  COMMON NO, NODE, NON, NOV, NP, NPAC
  COMMON NSAVE, NTL1,NU,NV,Q
  COMMON SPEED, TB, VHT, VLENG, VWD, VMT
  COMMON WDEL, WTM, WTHV, GIPFH, MDEL
  COMMON WDELCL, MDELCL, HRSINC
  COMMON MACARR,MACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
  1 (KR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  DO 200 IAC = 1, NACT
    J = 1
    DO 180 L = 1, NSTL
      DO 170 K = 1, 9
        IF (NP(K) - NPLG(L)) 170,175,170
      CONTINUE
      GO TO 180
    175 NPAC(IAC,J) = L
      J = J + 1
    180 CONTINUE
      NENT = J - 1
      IF (NENT - 1) 195,195,181
    181 NX = 0
      DO 190 KZ = 2, NENT
        L1 = NPAC(IAC,KZ - 1)
        L2 = NPAC(IAC,KZ)
        NO1 = NBASE(L1)
        NO2 = NBASE(L2)
        NPL1 = NAOTB(IAC,NO1)
        NPL2 = NAOTB(IAC,NO2)
        FLOW1 = WTHV(NPL1)
        FLOW2 = WTHV(NPL2)
        IF (FLOW1 - FLOW2) 185,190,190
    185 NSP = NPAC(IAC,KZ - 1)
        NPAC(IAC,KZ - 1) = NPAC(IAC,KZ)
        NPAC(IAC,KZ) = NSP
        NX = 1
      END DO
    END DO
  END DO

```

PINP0570
PINP0580
PINP0590
PINP0600
PINP0610
PINP0620
PINP0630
PINP0640

190 CONTINUE
191 IF (NX) 195,195,181
195 JNCW (IAC) = 1
196 IF (NENT - 50) 196,200,200
200 NPAC(IAC,NENT+1) = - 1
200 CONTINUE
201 RETURN
202 END

```

CPSAT      SUBROUTINE TO READ FIRST THREE CLASSES OF DATA
SUBROUTINE PRST(NVT,IVSP1,IVSP2,IVSP3,IVSP4,IVSP5,IVSP6)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANCL(10),APAY(10,5),ARR(1000)
  1,AMID(10),CL(50),DIST(50,50),EPAX(10),FLCH(50),GRT(50),JNCW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NACTB(10,50),NBASE(50),
  3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NRJUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NFARR(1000),NTL(50),NTL1(50),NU(7),NV(200),SEED(10),SPA(50)
  6,TB(50),TMIN(50),VHT(200),VLENG(200),VMD(200),VMT(200),WDEL(200),
  7,WFM(50),WTMV(200),XL(50),XR(50)
  8,WDEL(200),WDELCL(200),WDELCC(200)
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NFL,KL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AMID
  COMMON EPAX , GRT , JNCW , LISV , NAME , NAMEP
  COMMON NAMEV , NACTB , NBASE , NP , NPLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VMD , VMT
  COMMON WDEL , WFM , WTMV , WTPFH , WDEL
  COMMON WDELCL , WDELCC , HESINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRJUT(1)),(KL(1),CL(1)),
  1 (XR(1),TMIN(1)),(DIST(1),ARR(1)),(DIST(100),NTARR(1)),
  2 (DIST(200),NSTK(1))
C
C-->      ADD LABELED COMMON TO HOLD 5 MORE PRINT SWITCHES      6/12/75
COMMON /LAB01/ IVSP7,IVSP8,IVSP9,IVSP10,IVSP11
C
100  FORMAT (80H
1
101  FORMAT(I2,2X,A6,I5,3F5.0,F6.0,11I4)
102  FORMAT(19HC VEHICLE DATA )
103  FORMAT(119H0 NAME NUMBER WIDTH HEIGHT LENGTH HEIGHT UNPRST0380
1, 1 UN, 2 UN, 3 UN, 4 UN, 5 UN, 6 UN, 7 UN, 8 UN, 9 UN,10 UN,11 )
104  FORMAT(I2X,A6,I7,3F8.1,F10.1,2X,11I6)
105  FORMAT(I2,2X,I1,15I5)
106  FORMAT(8H0 LIST I1, 43H REJECTED - NUMBER MUST BE BETWEEN 1 ANDPRST0420
1 5 )
107  FORMAT(15H0 VEHICLE LIST I2,33H INCLUDES THE FOLLOWING VEHICLES )
108  FORMAT(1H 20I6 )
109  FORMAT(1H1 )
110  FORMAT(43H0280GRAM STOPPED - DATA CARDS OUT OF ORDER )
C
C-->      CHANGE FORMAT TO ALLOW FOR 11 PRINT SWITCHES      6/12/75
C-111  FORMAT(I2,3X,4F5.0,6I5)
111  FORMAT(I2,3X,4F5.0,11I5)
112  FORMAT(
112  REQUIREMENTS -- WIDTH - P5.1,11H HEIGHT - P5.1,11H LENGTH - P5.1PRST0530
2 )
C
190  READ (5,100,END=12345)
PRST0010
PRST0020
PRST0030
PRST0040
PRST0050
PRST0060
PRST0070
PRST0080
PRST0090
PRST0100
PRST0110
PRST0120
PRST0130
PRST0140
PRST0150
PRST0160
PRST0170
PRST0180
PRST0190
PRST0200
PRST0210
PRST0220
PRST0230
PRST0240
PRST0250
PRST0260
PRST0270
PRST0280
PRST0290
PRST0300
PRST0310
PRST0320
PRST0330
PRST0340
PRST0350
PRST0360
PRST0370
PRST0380
PRST0390
PRST0400
PRST0410
PRST0420
PRST0430
PRST0440
PRST0450
PRST0460
PRST0470
PRST0480
PRST0490
PRST0500
PRST0510
PRST0520
PRST0530
PRST0540
PRST0550
PRST0560

```



```

C-->
C
WRITE (6,109)
WRITE (6,100)
      ALLOW 11 PRINT SWITCHES ON CARD      6/12/75
      READ (5,111) ISW, CLW, CLH, CLL, HRSINC, IVSP1, IVSP2, IVSP3, IVSP4, IVSP5,
1 IVSP6, IVSP7, IVSP8, IVSP9, IVSP10, IVSP11
      IP (HRSINC) 230, 230, 235
      HRSINC = 12.0
230 IF (ISW) 240, 240, 999
235 IF (ISW) 240, 240, 999
240 WRITE (6,112) CLW, CLH, CLL
      READ (5,111) ISW
      IF (IVSP2) 260, 260, 250
250 WRITE (6,102)
      WRITE (6,103)
C READ VEHICLE DATA
260 DO 300 I = 1, 200
      READ (5,101) ISW, NAMEV(I), NV(I), VWD(I), VHT(I), VLENG(I), VWT(I),
1 V(I, J), J=1, 11)
      IF (ISW - 1) 350, 281, 350
281 IF (I - 1) 285, 279, 285
285 VWD(I) = VWD(I) + CLW
      VHT(I) = VHT(I) + CLH
      VLENG(I) = VLENG(I) + CLL
279 IF (IVSP2) 300, 300, 280
280 WRITE (6,104) NAMEV(I), NV(I), VWD(I), VHT(I), VLENG(I), VWT(I),
1 V(I, J), J=1, 11)
300 CONTINUE
      NVF = 200
      READ (5,101) ISW
      GO TO 400
350 NVF = I - 1
C READ VEHICLE LISTS
400 LNL = 0
      DO 404 I = 1, 5
      DO 404 J = 1, 100
      LISV(I, J) = 0
404 READ (5,105) ISW, LN, (NSAVE(I), I = 1, 15)
405 IF (ISW - 2) 450, 406, 450
406 IF (LN) 499, 499, 407
407 IF (LN - 5) 410, 410, 499
499 WRITE (6,106) LN
      GO TO 405
410 IF (LN - LNL) 420, 430, 420
420 DO 425 I = 1, 15
425 LISV(LN, I) = NSAVE(I)
      IB = 16
      LNL = LN
      GO TO 405
430 IF (IB - 91) 435, 436, 405
435 IN = 15
      GO TO 440
436 IN = 10
440 DO 445 I = 1, IN
      IX = IB + I - 1
445 LISV(LN, IX) = NSAVE(I)
      IB = IX + 1
      GO TO 405

```

PRST0570
 PRST0580
 PRST0590
 PRST0600
 PRST0610
 PRST0620
 PRST0630
 PRST0640
 PRST0650
 PRST0660
 PRST0670
 PRST0680
 PRST0690
 PRST0700
 PRST0710
 PRST0720
 PRST0730
 PRST0740
 PRST0750
 PRST0760
 PRST0770
 PRST0780
 PRST0790
 PRST0800
 PRST0810
 PRST0820
 PRST0830
 PRST0840
 PRST0850
 PRST0860
 PRST0870
 PRST0880
 PRST0890
 PRST0900
 PRST0910
 PRST0920
 PRST0930
 PRST0940
 PRST0950
 PRST0960
 PRST0970
 PRST0980
 PRST0990
 PRST1000
 PRST1010
 PRST1020
 PRST1030
 PRST1040
 PRST1050
 PRST1060
 PRST1070
 PRST1080
 PRST1090
 PRST1100
 PRST1110
 PRST1120
 PRST1130

```

C      PRINT VEHICLE LISTS
450  IF (IVSP2) 470,470,454
454  WRITE (6,109)
      DO 460 I = 1,5
      DO 451 J = 1, 100
      IF (LISV(I,J) ) 452,452,451
451  CONTINUE
      JMAX = 100
      GO TO 453
452  JMAX = J - 1
      IF (JMAX) 460,460,453
453  WRITE (6,107) I
      WRITE (6,108) (LISV(I,K) , K = 1,JMAX)
460  CONTINUE
C      CONVERT VEHICLE LISTS TO INTERNAL VEHICLE NUMBERS
470  DO 490 I = 1, 5
      IX = 1
      DO 480 J = 1,100
      IF (LISV(I,J) ) 482,482,475
475  DO 478 K = 1,200
      IF (NV(K) - LISV(I,J) ) 478,479,478
478  CONTINUE
      GO TO 480
479  LISV(I,IX) = K
      IX = IX + 1
480  CONTINUE
      GO TO 490
482  LISV(I,IX) = -1
490  CONTINUE
500  RETURN
999  WRITE (6,110)
12345 CALL EXIT
      RETURN
      END
PRST1140
PRST1150
PRST1160
PRST1170
PRST1180
PRST1190
PRST1200
PRST1210
PRST1220
PRST1230
PRST1240
PRST1250
PRST1260
PRST1270
PRST1280
PRST1290
PRST1300
PRST1310
PRST1320
PRST1330
PRST1340
PRST1350
PRST1360
PRST1370
PRST1380
PRST1390
PRST1400
PRST1410
PRST1420
PRST1430
PRST1440
PRST1450
PRST1460
PRST1470

```

```

CPSDL      SUBROUTINE TO COMPUTE FIRST DELIVERIES
SUBROUTINE FSDL (NVT,NNODES,NACT,NSTL,NACF,IWAR)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10,5),ARR(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLCH(50),GRT(50),JNOH(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NACTB(10,50),NBASE(50),
  3,NF(9),NFLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NECUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NFARR(1000),NTL(50),NTL1(50),NV(200),Q(7),SPEED(10),STA(50)
  6,TB(50),TAIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
  7,WTH(50),WTHV(2000),XL(50),XZ(50)
  8,WDEL(200),WDELCL(200),WDELCLC(200)
  DIMENSION NACARR(200),NACDEF(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AWID
  COMMON EPAX , GRT , JNOH , LISV , NAME , NAMEP
  COMMON NAMEV , NACTB , NBASE , NF , NFLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q , VHT , VLENG , VWD , VWT
  COMMON SPEED , TB , WTM , WTHV , WDEL , WDELCLC
  COMMON WDEL , WDELCL , WDELCLC , HESINC
  COMMON NACARR,NACDEF,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRQUT(1)),(XL(1),CL(1)),
  1 (KR(1),TAIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
C
C-->      ADD LABELED COMMON TO HOLD 5 MORE PRINT SWITCHES 6/12/75
COMMON /LAB01/ IVSP7,IVSP8,IVSP9,IVSP10,IVSP11
C
C-->      CHANGED THE FOLLOWING: TO REAL*8 5/29/75
REAL*8 AZ,NEZ,ERR
C
C      FORMAT (20H1 AIRCRAFT ARRIVALS )
C
C-->      FORMAT 101 NEVER REFERENCED. SAME AS 100 6/12/75
C-101  FORMAT(20H1 AIRCRAFT ARRIVALS )
102  FORMAT (79H0 BASE PLAS AIRCRAFT TYPE QUANTITY TIME BEFORE
1E FIRST TIME BETWEEN
103  FORMAT (1H0A6,2I4,4X,A6,4X,I13,F24.2,F16.2 )
104  FORMAT (12,3X,5I5,2F5.0 )
105  FORMAT(55H ABOVE AIRCRAFT REJECTED - BASE, A/C OR PLA; NOT INPUT )
106  FORMAT(50H0 PROGRAM STOPPED - MORE THAN 1000 AIRCRAFT INPUT )
107  FORMAT(1H 10X,I10,4H AIRCRAFT LOADED FROM LIST WITH FLAG NUMBER
1 I5 )
C
C-->      CHANGE FORMATS 108 109 TO DOUBLE SPACE 6/12/75
108  FORMAT (1H0, 10X, I10, 20H AIRCRAFT PLY EMPTY )
109  FORMAT (1H0, 15X,
1 49H ALL AIRCRAFT LOADED FROM LIST WITH FLAG NUMBER ,15)
C
C-->      110 & 111 FORMATS ADDED 6/12/75
110  FORMAT (/ / 1H ,29X,9H TOTAL /

```

FSDL0010
 FSDL0020
 FSDL0030
 FSDL0040
 FSDL0050
 FSDL0060
 FSDL0070
 FSDL0080
 FSDL0090
 FSDL0100
 FSDL0110
 FSDL0120
 FSDL0130
 FSDL0140
 FSDL0150
 FSDL0160
 FSDL0170
 FSDL0180
 FSDL0190
 FSDL0200
 FSDL0210
 FSDL0220
 FSDL0230
 FSDL0240
 FSDL0250
 FSDL0260
 FSDL0270
 FSDL0280
 FSDL0290
 FSDL0300
 FSDL0310
 FSDL0320
 FSDL0330
 FSDL0340
 FSDL0350
 FSDL0360
 FSDL0370
 FSDL0380
 FSDL0390
 FSDL0400
 FSDL0410
 FSDL0420
 FSDL0430
 FSDL0440
 FSDL0450
 FSDL0460
 FSDL0470
 FSDL0480
 FSDL0490
 FSDL0500
 FSDL0510
 FSDL0520
 FSDL0530
 FSDL0540
 FSDL0550
 FSDL0560


```

1 11X, 6HAIR ,3X, 6HONLOAD,3X, 9H WEIGHT,3X, 9H UNUSED ,3X,
2 9H CARGO,3X, 9HNUMBER OF /
3 11X, 6HCRAFT,3X, 6HBASE ,3X, 9H LOADED,3X, 9HCAPACITY ,3X,
2 9H WEIGHT,3X, 9HPERSONNEL / )
111 FORMAT (1H ,17,3X,A6,3X,A6,3F12.1,I12)
CALL ATHRUZ (Q000HL,6HERROR )
ERR = (+Q000HL)

C
C--> START NEW PAGE FOR EACH A/C 6/12/75
C--> WRITE (6,100)
C--> WRITE (6,102)
C--> NSW = 1
IX = 1
DO 120 IPG = 1,200
MDEL(IPG) = 0
WDEL(IPG) = 0.0
MDEL(IP3) = 0
WDEL(IP3) = 0
MACDER(IPG) = 0
MACARR(IP3) = 0
DO 130 IPG = 1,10
NORMAN(IPG) = 0
C
C--> SET FOR PRINTING DELIVERIES IF IVSP7 NOT ZERO 6/12/75
IF (IVSP7.EQ.0) ASSIGN 5450 TO KSWF
IF (IVSP7.NE.0) ASSIGN 5410 TO KSWF
C
140 READ (5,104)ISW,NBX,NFX1,NFX2,NA,NOTY,DELT,TBA
NSS = 0
IP (ISW - 9 ) 150,200,150
150 NSP = 0
NACP = IX - 1
DO 160 K = 2, NACP
IF (ARR(K-1) - ARR(K) ) 160,160,155
155 ASP = ARR(K-1)
ARR(K-1) = ARR(K)
ARR(K) = ASP
NSP = 1
NTSP = NTARR(K-1)
NTARR(K-1) = NTARR(K)
NTARR(K) = NTSP
CONTINUE
160 IF (NSP) 180,180,150
180 RETURN
C
C--> EJECT TO NEW PAGE FOR EACH A/C 6/12/75
C-200 IF (NBX) 140,140,201
200 WRITE (6,100)
201 WRITE (6,102)
DO 350 I = 1, NACT
ANA = NA
IF (ANA - ANO(I) ) 350,360,350
350 CONTINUE
AZ = ERR
C
C--> GET BASE NAME BEFORE PRINTING A/C ERROR 6/12/75
C--> GO TO 260

```

FSDL0570
 FSDL0580
 FSDL0590
 FSDL0600
 FSDL0610
 FSDL0620
 FSDL0630
 FSDL0640
 FSDL0650
 FSDL0660
 FSDL0670
 FSDL0680
 FSDL0690
 FSDL0700
 FSDL0710
 FSDL0720
 FSDL0730
 FSDL0740
 FSDL0750
 FSDL0760
 FSDL0770
 FSDL0780
 FSDL0790
 FSDL0800
 FSDL0810
 FSDL0820
 FSDL0830
 FSDL0840
 FSDL0850
 FSDL0860
 FSDL0870
 FSDL0880
 FSDL0890
 FSDL0900
 FSDL0910
 FSDL0920
 FSDL0930
 FSDL0940
 FSDL0950
 FSDL0960
 FSDL0970
 FSDL0980
 FSDL0990
 FSDL1000
 FSDL1010
 FSDL1020
 FSDL1030
 FSDL1040
 FSDL1050
 FSDL1060
 FSDL1070
 FSDL1080
 FSDL1090
 FSDL1100
 FSDL1110
 FSDL1120
 FSDL1130

```

360 GO TO 245
361 IAC = I
362 AZ = ANAME(IAC)
245 DO 250 I = 1, NNODES
363 IF (NO(I) - NBI) 250, 280, 250
250 CONTINUE
364 NBZ = NBR
260 WRITE (6, 103) NBZ, NFX1, NFX2, AZ, NQTY, DELT, IBA
365 WRITE (6, 105)
366 GO TO 140
280 NBX = I
367 NBZ = NAME(I)
C
C--> CHECK IF THERE WAS AN ERROR IN A/C INDEX 6/12/75
368 IF (AZ.EQ.ERR) GO TO 260
369 IF (NBI - IWAR) 439, 300, 439
300 WRITE (6, 103) NBZ, NFX1, NFX2, AZ, NQTY, DELT, IBA
370 DO 349 I = 1, NQTY
371 ZI = I - 1
372 ARR(IX) = DELT + (ZI * TBA) + GRT(IWAR)
373 NARR(IX) = IAC
374 NDT = (ARR(IX) / HRSINC) + 1.0
375 IF (NDT - 200) 340, 340, 335
335 NDI = 200
340 MACDEP(NDT) = MACDEP(NDT) + 1
341 NDFB = 200
342 NACARR(NDTB) = NACARR(NDTB) + 1
343 IF (IX - 1000) 345, 345, 570
345 IX = IX + 1
349 CONTINUE
376 GO TO 140
439 DO 450 I = 1, NSTL
440 IF (NBASE(I) - NBX) 450, 440, 450
450 IF (NPLG(I) - NFX1) 450, 470, 450
470 CONTINUE
470 LIST1 = I
471 DO 480 I = 1, NSTL
472 IF (NBASE(I) - NBX) 480, 475, 480
475 IF (NPLG(I) - NFX2) 480, 485, 480
480 CONTINUE
480 GO TO 260
485 LIST2 = I
486 NBEG = 0
487 LIST = LIST1
488 WRITE (6, 103) NBZ, NFX1, NFX2, AZ, NQTY, DELT, IBA
489 NOB = NBASE(LIST)
490 NJF = 0
491 NDS = 0
C
C--> SET SWITCH TO PRINT HEADINGS; STORE MAILLOAD 6/12/75
492 ASSIGN 5420 TO KSWH
493 WTMAX = APAY(IAC, 2)
494 DO 600 I = 1, NQTY

```

FSDL1140
FSDL1150
FSDL1160
FSDL1170
FSDL1180
FSDL1190
FSDL1200
FSDL1210
FSDL1220
FSDL1230
FSDL1240
FSDL1250
FSDL1260
FSDL1270
FSDL1280
FSDL1290
FSDL1300
FSDL1310
FSDL1320
FSDL1330
FSDL1340
FSDL1350
FSDL1360
FSDL1370
FSDL1380
FSDL1390
FSDL1400
FSDL1410
FSDL1420
FSDL1430
FSDL1440
FSDL1450
FSDL1460
FSDL1470
FSDL1480
FSDL1490
FSDL1500
FSDL1510
FSDL1520
FSDL1530
FSDL1540
FSDL1550
FSDL1560
FSDL1570
FSDL1580
FSDL1590
FSDL1600
FSDL1610
FSDL1620
FSDL1630
FSDL1640
FSDL1650
FSDL1660
FSDL1670
FSDL1680
FSDL1690
FSDL1700

FSDL2280
FSDL2290
FSDL2300
FSDL2310
FSDL2320
FSDL2330
FSDL2340
FSDL2350
FSDL2360

600 CONTINUE
IF (LIST - LIST1) 660,650,660
650 WRITE (6,109)NPK1
DO TO 140
660 IF (NDS) 140,661,140
661 NLDED = I-1-NBE;
WRITE (6,107)NLDED, NPK2
GO TO 140
END

```

CLOAD      SUBROUTINE TO LOAD AIRCRAFT
SUBROUTINE LOAD(IAC,PAY,LIST,NVT,WT,WPAK,NSIM,NINA)
DIMENSION AHT(10),ALENG(10),ANAME(10),ANQ(10),APAY(10),ARR(1000)
1.AWID(10),CL(50),DIST(50,50),EPAX(10),FLSH(50),GRT(50),JNOW(10),
LOAD0030
LOAD0040
LOAD0050
2.LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAMEB(10,50),NBASE(50),
LOAD0060
LOAD0070
LOAD0080
3.NF(9),NFLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
LOAD0090
LOAD0100
LOAD0110
4.NPAC(10,50),NRUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
LOAD0120
LOAD0130
LOAD0140
5.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0150
LOAD0160
LOAD0170
6.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0180
LOAD0190
LOAD0200
7.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0210
LOAD0220
LOAD0230
8.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0240
LOAD0250
LOAD0260
9.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0270
LOAD0280
LOAD0290
10.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0300
LOAD0310
LOAD0320
11.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0330
LOAD0340
LOAD0350
12.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0360
LOAD0370
LOAD0380
13.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0390
LOAD0400
LOAD0410
14.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0420
LOAD0430
LOAD0440
15.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0450
LOAD0460
LOAD0470
16.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0480
LOAD0490
LOAD0500
17.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0510
LOAD0520
LOAD0530
18.NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
LOAD0540
LOAD0550
LOAD0560

```



```

2102 PAYS = PAYS + STA(LINEX) - SL
C
258 IR(LINEX) = AVID(IAC)
C
C JOB COMPLETED, SEARCH FOR A NEW GAP
GO TO 350
C
C SPACE IS BETWEEN TWO LINES - LINE AND LINEL
C
260 IF ( STA(LINE) - STA(LINEL) ) 261, 261, 265
C
C LINE IS CLOSER, EXTEND IT
261 XL(LINE) = IR(LINEL)
C
C JOB COMPLETED, SEARCH FOR A NEW GAP
GO TO 350
C
C LINEL IS CLOSER, EXTEND IT
265 IR(LINEL) = XL(LINE)
C
C JOB COMPLETED, SEARCH FOR A NEW GAP
GO TO 350
C
C VEHICLE FOUND, ADD ITS WEIGHT AND SPACE TO POPAL LOAD
396 WT = WT + VWT(IBEST)
C
C ADJUST NUMBER LEFT
MSK(LIST,IBEST) = MSK(LIST,IBEST) - 1
MSK(16,IBEST) = MSK(16,IBEST) - 1
C
C RECORD VEHICLE IN NSAVE
IF (NINA - 50) 3101,3100,3100
3100 NINA = 51
GO TO 3102
3101 NINA = NINA + 1
NSAVE(NINA) = IBEST
C
C INSERT A NEW LINE IN NTL TO MARK THE END OF THE VEHICLE
3102 DO 397 KNEW = 1, 50
IF ( STA (KNEW) ) 398, 398, 397
397 CONTINUE
GO TO 390
398 XL(KNEW) = IF
IR(KNEW) = IF + VWD(IBEST)
STA(KNEW) = VLEN3(IBEST) + SL
C
C LINE FOUND AND VALUES ASSIGNED, NOW INSERT IN NTL
IF (NIN) 401, 401, 405
401 NIN = 1
NTL(NIN) = KNEW
GO TO 439
405 DO 420 KZ = 1, NIN
LINE = NTL(KZ)
IF ( XL(KNEW) - XL(LINE) ) 430, 420, 420
420 CONTINUE
NIN = NIN + 1
NTL(NIN) = KNEW
GO TO 439
430 KZP1 = KZ + 1
NIN = NIN + 1
DO 435 KQ = KZP1, NIN
435 NTL(KQ) = NTL(KQ - 1)
DO 436 KQ = KZP1, NIN
436 NTL(KQ) = NTL(KQ)
NTL(KZ) = KNEW
439 GO TO 350
C
C NO GAP EXISTS, RESET SL

```

LOAD1180
 LOAD1150
 LOAD1160
 LOAD1170
 LOAD1180
 LOAD1190
 LOAD1200
 LOAD1210
 LOAD1220
 LOAD1230
 LOAD1240
 LOAD1250
 LOAD1260
 LOAD1270
 LOAD1280
 LOAD1290
 LOAD1300
 LOAD1310
 LOAD1320
 LOAD1330
 LOAD1340
 LOAD1350
 LOAD1360
 LOAD1370
 LOAD1380
 LOAD1390
 LOAD1400
 LOAD1410
 LOAD1420
 LOAD1430
 LOAD1440
 LOAD1450
 LOAD1460
 LOAD1470
 LOAD1480
 LOAD1490
 LOAD1500
 LOAD1510
 LOAD1520
 LOAD1530
 LOAD1540
 LOAD1550
 LOAD1560
 LOAD1570
 LOAD1580
 LOAD1590
 LOAD1600
 LOAD1610
 LOAD1620
 LOAD1630
 LOAD1640
 LOAD1650
 LOAD1660
 LOAD1670
 LOAD1680
 LOAD1690
 LOAD1700

```

390 SL = ALENG(IAC)
DO 392 K = 1, NIN
LINEZ = NTL(K)
IF ( STA(LINEZ) - SL ) 391, 391, 392
C LINE IS FORWARD OR ABEAST OF PRESENT SL
C RESET SL
391 SL = STA(LINEZ)
392 CONTINUE
C SL IS NOW RESET
C REMOVE ALL TAIL LINES ALONG SL
C SECTION TO REMOVE ALL TAIL LINES WITH STA(I) = SL FROM NTL VECTOR
450 NREHV = 0
C SET NTL TO ZERO FOR LINES TO BE REMOVED
DO 460 KZ = 1, NIN
LINE = NTL(KZ)
IF ( STA(LINE) - SL ) 460, 455, 460
455 NREHV = NREHV + 1
460 STA(LINE) = 0.0
CONTINUE
NFOUND = 0
NINOW = NIN
DO 458 KQ = 1, NIN
IF ( NTL(KQ) ) 451, 451, 458
451 NZER = 0
DO 452 KJB = KQ, NINOW
IF ( NTL(KJB) ) 462, 462, 463
462 NZER = NZER + 1
452 CONTINUE
463 NFOUND = NFOUND + NZER
NINOW = NIN - NFOUND
DO 453 KM = KQ, NINOW
KIRO = KM + NZER
453 NTL(KM) = NTL ( KIRO )
IF ( NFOUND - NREHV ) 458, 461, 461
458 CONTINUE
461 NIN = NIN - NREHV
C ALL ADJUSTMENTS MADE - RETURN TO FIND A NEW GAP
GO TO 350
C NO VEHICLE WAS FOUND AT A TIME WHEN NO LINES WERE IN, THUS NO
C MORE CAN BE LOADED IN THIS AIRCRAFT
C SEE IF THIS IS AN EMPTY AIRCRAFT
C NO MORE CAN BE LOADED IN THIS AIRCRAFT
500 IF (SL) 600, 600, 501
C NO VEHICLES ABOARD
600 NPAXS = EPAX(IAC)
LAB = 1
GO TO 2314
501 PAXS = PAXS + 2.0* (ALENG(IAC) - SL )
NPAXS = PAXS / VLENG(1)
2314 NPAXW = (PAX - WT) / VWT(1)
IF (NPAXS - NPAXW) 2315, 2315, 2320
2315 NPAX = NPAXS
GO TO 2328
2320 NPAX = NPAXW
2328 IF (NPAX - NSTK(LIST,1) ) 2330, 2330, 2329
2329 NPAX = NSTK (LIST,1)

```

LOAD1710
 LOAD1720
 LOAD1730
 LOAD1740
 LOAD1750
 LOAD1760
 LOAD1770
 LOAD1780
 LOAD1790
 LOAD1800
 LOAD1810
 LOAD1820
 LOAD1830
 LOAD1840
 LOAD1850
 LOAD1860
 LOAD1870
 LOAD1880
 LOAD1890
 LOAD1900
 LOAD1910
 LOAD1920
 LOAD1930
 LOAD1940
 LOAD1950
 LOAD1960
 LOAD1970
 LOAD1980
 LOAD1990
 LOAD2000
 LOAD2010
 LOAD2020
 LOAD2030
 LOAD2040
 LOAD2050
 LOAD2060
 LOAD2070
 LOAD2080
 LOAD2090
 LOAD2100
 LOAD2110
 LOAD2120
 LOAD2130
 LOAD2140
 LOAD2150
 LOAD2160
 LOAD2170
 LOAD2180
 LOAD2190
 LOAD2200
 LOAD2210
 LOAD2220
 LOAD2230
 LOAD2240
 LOAD2250
 LOAD2260
 LOAD2270

```

2330 I P(NPAX - NSTK(16,1)) 2340,2340,2335
2335 NPAX = NSTK(16,1)
2340 IF (NPAX) 2342,2342,2341
2341 NSTK(LIST,1) = NSTK(LIST,1) - NPAX
      NSTK(16,1) = NSTK(16,1) - NPAX
      WT = WT + FLOAT(NPAX) * VWT(1)
860 IF (NSTK(LIST,1)/NPAX - NSTK(16,1)/NPAX) 865,865,870
865 NPAIR = NSTK(LIST,1)/NPAX
      GO TO 2342
870 NPAXR = NSTK(16,1) / NPAX
2342 IF (LAW) 3000, 3000, 4102
3000 NMIN = 1000000
      IF (NMIN - 50) 910, 910, 850
910 DO 950 K1 = 1, NMIN
      NENT = 1
      K1P1 = K1 + 1
      DO 940 K2 = K1P1, NMIN
      IF (NSAVE(K2) - NSAVE(K1)) 940,941,940
941 NENT = NENT + 1
940 CONTINUE
      KSP = NSAVE(K1)
      NSP = NSTK(LIST,KSP) / NENT
      IF (NSP - NMIN) 945,945,946
945 NMIN = NSP
946 NSP = NSTK(16,KSP) / NENT
      IF (NSP - NMIN) 947,947,950
947 NMIN = NSP
950 CONTINUE
960 IF (NPAX) 980,980,965
965 IF (NPAXR - NMIN) 972,972,980
4102 LAW = 0
      IF (NPAX) 971, 971, 972
850 NSIM = 0
      RETURN
971 NSIM = 1000000
      RETURN
972 NSIM = NPAXR
      RETURN
980 NSIM = NMIN
      RETURN
      END
LOAD2280
LOAD2290
LOAD2300
LOAD2310
LOAD2320
LOAD2330
LOAD2340
LOAD2350
LOAD2360
LOAD2370
LOAD2380
LOAD2390
LOAD2400
LOAD2410
LOAD2420
LOAD2430
LOAD2440
LOAD2450
LOAD2460
LOAD2470
LOAD2480
LOAD2490
LOAD2500
LOAD2510
LOAD2520
LOAD2530
LOAD2540
LOAD2550
LOAD2560
LOAD2570
LOAD2580
LOAD2590
LOAD2600
LOAD2610
LOAD2620
LOAD2630
LOAD2640
LOAD2650
LOAD2660
LOAD2670
LOAD2680

```



```

CMAIN SUBROUTINE TO FIND MINIMUM TIME THROUGH A NETWORK
SUBROUTINE MINT (CLMAX, SPEEDY, WINDP, THINK, T3DEST, CLX)
DIMENSION AHT (10), ALENG (10), ANAME (10), ANJ (10), APAY (10, 5), ARR (1000)
1. AVID (10), CL (50), DIST (50, 50), EPAY (10), FLOW (50), GRI (50), INCM (10),
2. LISV (5, 100), NAME (50), NAMEP (50), NAMEV (200), NAOTB (10, 50), NBASE (50),
3. NP (9), NPLG (50), NL (7), NO (50), NODE (50), NON (50), NOV (200, 11), NP (50),
4. NPAC (10, 50), NROUT (50), NSAVE (50), NSTK (16, 200), NSUBM (50, 50),
5. NIARR (1000), NTL (50), NTL1 (50), NU (7), NV (200), Q (7), SPEED (10), SKA (50),
6. T3 (50), THINK (50), VHT (200), VLEN (200), VWD (200), VWT (200), WDEL (200),
7. WTH (50), WTV (2000), XL (50), XR (50)
8. WDEL (200), WDELCL (200), WDELCL (200)
DIMENSION MACARR (200), MACDEP (200), NORMAN (10)
DIMENSION FILL (250)
REAL*8 ANAME, NAME, NAMEV
COMMON STA, NTL, XL, XR, DIST, NSUBM
COMMON FILL
COMMON AHT, ALENG, ANAME, ANO, ARAY, AVID
COMMON EPAY, GRI, JNOW, LISV, NAME, NAMEP
COMMON NAMEV, NAOTB, NBASE, NF, NPLG, NL
COMMON NO, NODE, NON, NOV, NP, NPAC
COMMON NSAVE, NTL1, NU, NV, Q
COMMON SPEED, TB, VHT, VLENG, VWD, VWT
COMMON WDEL, RTM, RTMV, GTPFH, WDEL
COMMON WDELCL, WDELCL, RESINC
COMMON MACARR, MACDEP, NORMAN
EQUIVALENCE (STA(1), FLOW(1)), (NTL(1), NROUT(1)), (XL(1), CL(1)),
1. (XR(1), THINK(1)), (DIST(1), ARR(1)), (DIST(1001), NTARR(1)),
2. (DIST(2002), NSTK(1))
DO 150 I = 1, 50
IF (NODE(I)) 160, 150, 150
150 CONTINUE
NEW = 50
GO TO 200
160 NEW = I - 1
C INITIALIZE VECTORS
200 N1 = NODE(I)
THIN (N1) = 0.0
DO 250 I = 2, NEW
ND = NODE(I)
THIN (ND) = 999999.0
CL (ND) = 0.0
NP (ND) = 0
CL (N1) = 0.0
C FIND ROUTES
250 NEW = 0
DO 500 J = 2, NEW
JN1 = J - 1
NFR = NODE(I)
NTO = NODE(J)
IF (NFR - NTO) 309, 301, 301
301 IF (DIST (WTO, NFR) - 999999.0) 302, 400, 400
302 DEQ = DIST (WTO, NFR)
GO TO 311
309 IF (DIST (NFR, NTO) - 999999.0) 310, 400, 400

```

MINT0010
MINT0020
MINT0030
MINT0040
MINT0050
MINT0060
MINT0070
MINT0080
MINT0090
MINT0100
MINT0110
MINT0120
MINT0130
MINT0140
MINT0150
MINT0160
MINT0170
MINT0180
MINT0190
MINT0200
MINT0210
MINT0220
MINT0230
MINT0240
MINT0250
MINT0260
MINT0270
MINT0280
MINT0290
MINT0300
MINT0310
MINT0320
MINT0330
MINT0340
MINT0350
MINT0360
MINT0370
MINT0380
MINT0390
MINT0400
MINT0410
MINT0420
MINT0430
MINT0440
MINT0450
MINT0460
MINT0470
MINT0480
MINT0490
MINT0500
MINT0510
MINT0520
MINT0530
MINT0540
MINT0550
MINT0560

```

310 DEQ = DIST(NPR,NTO)
311 IP (DEQ - CLMAX) 320,320,400
320 TI = (DEQ/SPEEDX) * (1.0 + GTPH) + GRT(NTO) + TMIN(NPR)
330 IF (TI - THIN(NTO)) 330,400,400
330 THIN(NTO) = TI
NP(NTO) = NPR
NSW = 1
IP (DEQ - CL(NPR)) 335,340,340
335 CL(NTO) = CL(NPR)
GO TO 400
340 CL(NTO) = DEQ
400 CONTINUE
500 CONTINUE
510 IP (NSW) 510,510,260
NLQ = NODE(NEN)
THINX = THIN(NLQ)
CLX = CL(NLQ)
NPI = NP(NLQ)
IF (NPI - NLQ) 502,503,503
502 F3DEST = ((JTPPH * DIST(NLQ,NPI)) / (SPEEDX + DIST(NLQ,NPI))) + JBT(NLQ)
GO TO 600
503 F3DEST = ((JTPPH * DIST(NLQ,NPI)) / (SPEEDX + DIST(NLQ,NPI))) + JBT(NLQ)
600 RETURN
END

```

MINT0570
 MINT0580
 MINT0590
 MINT0600
 MINT0610
 MINT0620
 MINT0630
 MINT0640
 MINT0650
 MINT0660
 MINT0670
 MINT0680
 MINT0690
 MINT0700
 MINT0710
 MINT0720
 MINT0730
 MINT0740
 MINT0750
 MINT0760
 MINT0770
 MINT0780
 MINT0790
 MINT0800

```

CNETH      SUBROUTINE TO PERFORM NETWORK ANALYSIS
SUBROUTINE NETW (IWAR, NACT, NNODES, IVSP1)
  DIMENSION AHT(10), ALENG(10), ANAME(10), ANO(10), APAY(10,5), ARR(1000)
  1, AVID(10), CL(50), DIST(50,50), EFAX(10), FLOW(50), GRT(50), JNOW(10),
  2 LISV(5,100), NAME(50), NAMEP(50), NAMEV(200), NACTB(10,50), NBASE(50),
  3 NP(9), NPL3(50), NL(7), NO(50), NODE(50), NON(50), NOV(200,11), NP(50),
  4 NPAC(10,50), NRUT(50), NSAVE(50), NSIK(15,200), NSUBN(50,50),
  5 NTAER(1000), NTL(50), NTL1(50), NV(200), Q(7), SPEED(10), STA(50)
  6 TB(50), TMIN(50), VHT(200), VLENG(200), VWD(200), VWT(200), WDEL(200),
  7 WTH(50), WTHV(2000), XL(50), XR(50)
  8 WDEL(200), WDEL2(200), WDEL3(200), NORMAN(10)
  DIMENSION NACARR(200), NACDEP(200), NAMEV
  DIMENSION FILL(250)
  REAL*8 ANAME, NAMEP
  COMMON STA, NTL, XL, KR, DIST, NSUBN
  COMMON FILL
  COMMON AHT, ALENG, ANAME, ANO, APAY, AVID
  COMMON EPAX, GRT, JNOW, LISV, NAME, NAMEP
  COMMON NAMEV, NACTB, NBASE, NF, NPLG, NL
  COMMON NO, NODE, NON, NOV, NP, NPAC
  COMMON NSAVE, NTL1, NV, NV, Q, VHT, VLENG, VWD, VWT
  COMMON SPEED, TB, WTH, WTHV, GTPPH, WDEL
  COMMON WDEL, WDEL2, WDEL3, HESINC
  COMMON NACARR, NACDEP, NORMAN
  EQUIVALENCE (STA(1), FLOW(1)), (NTL(1), NRUT(1)), (XL(1), CL(1)),
  1 (KR(1), TMIN(1)), (DIST(1), AFR(1)), (DIST(1001), NTAB(1)),
  2 (DIST(2002), NSIK(1))
  99 FORMAT (43HOPROGRAM STOPPED - DATA CARDS OUT OF ORDER )
  100 FORMAT (13H1 ROUTES FOR A6, 9H AIRCRAFT )
  101 FORMAT (100HOPROGRAM STOPPED TOO MANY ENTRIES IN WTHV - REDUCE NUMBER)
  102 FORMAT (12,2X,A6,F5.0,8F5.0,2F8.0,F7.0,12)
  103 FORMAT (16H0 AIRCRAFT DATA )
  104 FORMAT(21H COMPARTMENT IS F8.1,7H WIDE, F8.1, 10H HIGH AND
  105 F8.1, 7H LONG, F6.0, 12H PASSENGERS)
  15 F10.1, 12H RANGES ARE 3P9.1 )
  106 FORMAT (1H )
  107 FORMAT (12,3X, 15I5 )
  108 FORMAT (9H TIME IS F10.2,14H PAYLOAD IS F10.1 )
  109 FORMAT (46HXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX )
  110 FORMAT (46H0 PROGRAM STOPPED - WAR BASE NOT IN BASE LIST )
  111 FORMAT (15H SPEED IS F8.0, 9H GTPPH= F8.2)
  C
  C--> NEW FORMAT STATEMENT ADDED 6/12/75
  112 FORMAT (1H0, 12HONLOAD BASE, I4,
  1 25H IS NOT IN LIST OF BASES /)
  CALL RDNT (NNODES)
  CALL NWLK (NNODES, IVSP1)
  WRITE (6,106)
  250 IB = 1
  IBP = IB + 13
  IX = 1
  READ (5,107) ISW, IWAR, (NON(I), I = IB, IBP )

```

```

NETW0010
NETW0020
NETW0030
NETW0040
NETW0050
NETW0060
NETW0070
NETW0080
NETW0090
NETW0100
NETW0110
NETW0120
NETW0130
NETW0140
NETW0150
NETW0160
NETW0170
NETW0180
NETW0190
NETW0200
NETW0210
NETW0220
NETW0230
NETW0240
NETW0250
NETW0260
NETW0270
NETW0280
NETW0290
NETW0300
NETW0310
NETW0320
NETW0330
NETW0340
NETW0350
NETW0360
NETW0370
NETW0380
NETW0390
NETW0400
NETW0410
NETW0420
NETW0430
NETW0440
NETW0450
NETW0460
NETW0470
NETW0480
NETW0490
NETW0500
NETW0510
NETW0520
NETW0530
NETW0540
NETW0550
NETW0560

```



```

251 IF (ISW - 5) 999,246,999
246 DO 247 J = 1, NNODES
247 IF (NO(J) - IWAR) 247,248,247
    CONTINUE
    WRITE (6,110)
    CALL EXIT
248 IWAR = J
252 DO 258 I = IB, IBP
253 IF (NON(I)) 259,259,253
254 DO 254 J = 1, NNODES
255 IF (NO(J) - NON(I)) 254,255,254
256 CONTINUE
C
C--> PRINT MESSAGE IF ONLOAD BASE IS NOT IN INPUT SET
    WRITE (6,112) NON(I)
    GO TO 258
255 NON(IX) = J
IX = IX + 1
258 CONTINUE
IB = IB + 14
IBP = IB + 13
C
C--> CHECK FOR LIMIT ON TOTAL BASES 6/12/75
    IF (IBP.GT.50) IBP = 50
    READ (5,107) ISW,IZ, (NON(I), I=IB,IBP)
C
C--> IF NUMBER OF ONLOAD BASES IS A MULTIPLE OF 14, CARD WILL
C--> NOT TERMINATE WITH ZERO OR BLANK AS EXPECTED, THE NEXT CARD
C--> SHOULD BE BLANK. 6/12/75
    IF (ISW.EQ.0) GO TO 260
    IF (ISW - 5) 999,252,260
    IF (ISW - 5) 999,252,999
260 NON(IX) = - 1
    GO TO 280
259 NON(IX) = - 1
    READ (5,107) ISW
    IF (ISW - 5) 280,999,280
280 IT = 1
    DO 500 IAC = 1, 10
    READ (5,102) ISW, ANAME(IAC), ANO(IAC), AWID(IAC), AHT(IAC), ALENG(IAC)
    1,EPAY(IAC),SPEED(IAC),APAY(IAC,1),APAY(IAC,3),APAY(IAC,5),APAY(IAC,NET0970
    2,2),APAY(IAC,4),GTPPH,NBDN
    IF (ISW - 6) 210,190,210
    IF (ISW - 6) 210,190,210
190 WRITE (6,100) ANAME(IAC)
    WRITE (6,104) AWID(IAC),AHT(IAC),ALENG(IAC),EPAY(IAC)
    WRITE (6,105) APAY(IAC,2),APAY(IAC,4),APAY(IAC,1),APAY(IAC,3),APAY(IAC,5)
    1(IAC,5)
    WRITE (6,111) SPEED(IAC), GTPPH
    CALL SUBN (IAC, IWAR, NNODES,NBDN)
    DO 400 I = 1,50
    IF (NON(I)) 500, 500, 310
310 NOL = NON(I)
    CALL SBN2 (NOL, IWAR )
    CALL WTIN(IAC)
    NACTB(IAC, NOL) = IT
    DO 340 J = 1, 50
    IF (WTH(J) ) 350,340,340

```

NET0570
 NET0580
 NET0590
 NET0600
 NET0610
 NET0620
 NET0630
 NET0640
 NET0650
 NET0660
 NET0670
 NET0680
 NET0690
 NET0700
 NET0710
 NET0720
 NET0730
 NET0740
 NET0750
 NET0760
 NET0770
 NET0780
 NET0790
 NET0800
 NET0810
 NET0820
 NET0830
 NET0840
 NET0850
 NET0860
 NET0870
 NET0880
 NET0890
 NET0900
 NET0910
 NET0920
 NET0930
 NET0940
 NET0950
 NET0960
 NET0970
 NET0980
 NET0990
 NET1000
 NET1010
 NET1020
 NET1030
 NET1040
 NET1050
 NET1060
 NET1070
 NET1080
 NET1090
 NET1100
 NET1110
 NET1120
 NET1130

```

340 CONTINUE
    NCT = 50
    GO TO 355
350 NCT = J - 1
355 WTHV(IT) = WTH(NCT-1)/(WTH(NCT) + TB(NCL))
    IT = IT + 1
    WRITE (6,108) WTH(NCT), WTH(NCT - 1)
    WRITE (6,109)
    DO 360 K = 1, NCT
    KZ = NCT + 1 - K
    WTHV(IT) = WTH(KZ)
    IT = IT + 1
360 WTHV(IT-1) = - WTHV(IT-1)
    IF (IT - 2000) 400,400,600
600 WRITE (6,101)
    CALL EXIT
400 CONTINUE
500 CONTINUE
    NACT = 10
    READ (5,102) ISW
    IF (ISW - 6) 550,999,550
210 NACT = IAC - 1
550 RETURN
999 WRITE (6,99)
    CALL EXIT
    RETURN
    END

```

```

NETW1140
NETW1150
NETW1160
NETW1170
NETW1180
NETW1190
NETW1200
NETW1210
NETW1220
NETW1230
NETW1240
NETW1250
NETW1260
NETW1270
NETW1280
NETW1290
NETW1300
NETW1310
NETW1320
NETW1330
NETW1340
NETW1350
NETW1360
NETW1370
NETW1380
NETW1390
NETW1400

```

```

CWLK      SUBROUTINE TO ADD NEW NON-STOP LINKS TO DATA
          SUBROUTINE NWLKNODES,IVSP1)
          DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10,5),ARR(1000),
1        AVID(10),CL(50),DIST(50,50),EPAY(10),FLSH(50),GRT(50),JNOW(10),
2        LISV(5,100),NAME(50),NAMEP(50),NAMEY(200),NACB(10,50),NBASE(50),
3        NPAC(10,50),NPLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,1),NP(50),
4        NPAC(10,50),NRUT(50),NSAVE(50),NSTK(16,200),NSUBH(50,50),
5        NFARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50),
6        TB(50),THIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
7        WTM(50),WTMV(2000),XL(50),XR(50),
8        WDEL(200),WDELCL(200),WDELCL(200)
          DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
          DIMENSION FILL(250)
          REAL*8 ANAME,NAME,NAMEY
          COMMON STA,NTL,XL,XR,DIST,NSUBH
          COMMON FILL
          COMMON AHT , ALENG , ANAME , ANO , APAY , AVID
          COMMON EPAY , GRT , JNOW , LISV , NAME , NAMEP
          COMMON NAMEY , NACB , NBASE , NP , NPLG , NL
          COMMON NO , NODE , NON , NOV , NP , NPAC
          COMMON NSAVE , NTL1,NU,NV,Q
          COMMON SPEED , TB , VHT , VLENG , VWD , VWT
          COMMON WDEL , WTM , WTMV , WTPPH , WDEL
          COMMON WDELCL , WDELCL , HRSINC
          COMMON NACARR,NACDEP,NORMAN
          EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRUT(1)),(XL(1),CL(1)),
1        (KR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
2        (DIST(2002),NSTK(1))
115      FORMAT (40H1 TABLE OF SHORTEST ALLOWABLE DISTANCES
116      )
117      FORMAT ( 8(3X,A6,F7.0))
          NH1 = NNODES - 1
          DO 500 I = 1, NH1
            IP1 = I + 1
            DO 150 H = 1, NNODES
              CL(I) = 0.0
              THIN(I) = 0.0
              NSW = 0
              DO 300 J = 1, NNODES
                DO 250 K = 1, NNODES
                  IF (J - K) 241, 250, 242
241              DX = THIN(K) + DIST(J,K)
242              DI = THIN(K) + DIST(K,J)
243              IF (DX - THIN(J)) 245, 250, 250
245              THIN(J) = DI
              NSW = 1
250              CONTINUE
300              CONTINUE
400              IF (NSW) 400, 400, 160
401              IF (IP1 - NNODES) 401, 401, 550
402              DO 450 H = IP1, NNODES
450              DIST(I,H) = THIN(H)
              CONTINUE

```

NWLK0010
NWLK0020
NWLK0030
NWLK0040
NWLK0050
NWLK0060
NWLK0070
NWLK0080
NWLK0090
NWLK0100
NWLK0110
NWLK0120
NWLK0130
NWLK0140
NWLK0150
NWLK0160
NWLK0170
NWLK0180
NWLK0190
NWLK0200
NWLK0210
NWLK0220
NWLK0230
NWLK0240
NWLK0250
NWLK0260
NWLK0270
NWLK0280
NWLK0290
NWLK0300
NWLK0310
NWLK0320
NWLK0330
NWLK0340
NWLK0350
NWLK0360
NWLK0370
NWLK0380
NWLK0390
NWLK0400
NWLK0410
NWLK0420
NWLK0430
NWLK0440
NWLK0450
NWLK0460
NWLK0470
NWLK0480
NWLK0490
NWLK0500
NWLK0510
NWLK0520
NWLK0530
NWLK0540
NWLK0550
NWLK0560

NWLK0570
NWLK0580
NWLK0590
NWLK0600
NWLK0610
NWLK0620
NWLK0630
NWLK0640
NWLK0650
NWLK0660

```
500 CONTINUE
    IF (IVSP1) 550,550,501
501  WRITE (6,115)
     DO 540 I = 1,NM1
     WRITE (6,116) NAME(I)
     IP1 = I + 1
     WRITE (6,117) (NAME(J),DIST(I,J),J=IP1,NMODES)
540  CONTINUE
550  RETURN
     END
```

```

CPAYL      FUNCTION TO COMPUTE MAXIMUM PAYLOAD
          FUNCTION PAYL(I,DX)
          DIMENSION AHT(10),ALENG(10),ANAME(10),ANCI(10),APAY(10,5),ARR(1000)
          1,AWID(10),CL(50),DIST(50,50),EPAX(10),FLOH(50),GRT(50),JNOH(10),
          2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NACDEP(10,50),NBASE(50),
          3,NF(9),NPLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
          4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBM(50,50),
          5,NTABR(1000),NVL(50),NTL(50),NU(7),NV(206),Q(7),SPEED(10),STA(50)
          6,TB(50),TMIN(50),VHT(200),VLENG(200),VND(200),VNT(200),WDEL(200),
          7,WTM(50),WTMV(2000),XL(50),XR(50)
          8,WDEL(200),WDELC(200),WDELC(200)
          DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
          DIMENSION FILL(250)
          REAL*8 ANAME,NAME,NAMEV
          COMMON STA,NTL,XL,XR,DIST,MSUBM
          COMMON FILL
          COMMON AHT , ALENG , ANAME , ANCI , APAY , AWID
          COMMON EPAX , GRT , JNOH , LISV , NAME , NAMEP
          COMMON NAMEV , NACDEP , NBASE , NP , NPLG , NL
          COMMON NO , NODE , NON , NOV , NP , NPAC
          COMMON NSAVE , NTL,NU,NV,Q
          COMMON SPEED , TB , VHT , VLENG , VND , VNT
          COMMON WDEL , WTM , WTMV , STPPH , WDEL
          COMMON WDELC , WDELC , HRSINC
          COMMON NACARR,NACDEP,NORMAN
          EQUIVALENCE (STA(1),FLOH(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
          1 (IR(1),TMIN(1)), (DIST(1),ARR(1)), (DIST(1001),NTARR(1)),
          2 (DIST(2002),NSTK(1))
          IF (DI - APAY(I,1) ) 200,200,250
          200 PAYL = APAY(I,2)
          GO TO 500
          250 IF (DX - APAY(I,3) ) 300,300,350
          300 PAYL=APAY(I,4)+ (APAY(I,3)-DX )*(APAY(I,2)-APAY(I,4))/(APAY(I,3)
          1 - APAY(I,1) )
          30 TO 500
          350 IF (DX - APAY(I,5) ) 400,400,450
          400 PAYL = (( APAY(I,5) - DX ) * APAY(I,4))/(APAY(I,5)-APAY(I,3))
          GO TO 500
          450 PAYL = 0.0
          500 RETURN
          END
PAYL0010
PAYL0020
PAYL0030
PAYL0040
PAYL0050
PAYL0060
PAYL0070
PAYL0080
PAYL0090
PAYL0100
PAYL0110
PAYL0120
PAYL0130
PAYL0140
PAYL0150
PAYL0160
PAYL0170
PAYL0180
PAYL0190
PAYL0200
PAYL0210
PAYL0220
PAYL0230
PAYL0240
PAYL0250
PAYL0260
PAYL0270
PAYL0280
PAYL0290
PAYL0300
PAYL0310
PAYL0320
PAYL0330
PAYL0340
PAYL0350
PAYL0360
PAYL0370
PAYL0380
PAYL0390
PAYL0400
PAYL0410
PAYL0420

```

```

CPCDL      SUBROUTINE TO PRINT CUMULATIVE DELIVERIES
SUBROUTINE PCDEL(VMT,WTCT,MPAT,POUT,MACF,IVSR4,IVSR6,MACT)
  DIMENSION AHT(10),ALENG(10),ANAME(10),APAY(10),ARR(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAY(10),FLOW(50),GRT(50),JROW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAMEB(10,50),NAMESE(50),
  3,NPAC(10,50),NEOUT(50),NODE(50),NOM(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NEOUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NPAC(1000),NVL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
  6,TA(50),THIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
  7,WFM(50),WTMV(2000),XL(50),XR(50)
  8,WDEL(200),WDELCL(200),WDELCLC(200)
  DIMENSION POUT(51)
  DIMENSION MACARR(200),MACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AWID
  COMMON EPAY , GRT , JROW , LISV , NAME , NAMEP
  COMMON NAMEV , NAMEB , NAMESE , NP , NAME , NL
  COMMON NO , NODE , NOM , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VWD , VWT
  COMMON WDEL , WTR , WTV , WTPH , WDEL
  COMMON WDELCL , WDELCLC , HRSINC
  COMMON MACARR,MACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NEOUT(1)),(XL(1),CL(1)),
  1 (XR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  100 FORMAT( 90H1 CUMULATIVE DELIVERIES OF MATERIAL AND PERSONNEL FROM
  1ALL PRIORITY GROUPS TO OFFLOAD BASE )
  101 FORMAT(64H0 PERSONNEL PERCENT OF PERSONNEL PERCENT OF CARGO
  1AY HOUR
  102 FORMAT(1H 10, F20.2, F21.2, I7, I5, 1X, 51A1)
  106 FORMAT(1H 64X, 51H-----)
  1----
  108 FORMAT(1H0 F10.2, I7)
  109 FORMAT(24H1 OFFLOAD BASE ACTIVITY )
  110 FORMAT(81H0 **FROM*** **TO**
  1 AIRCRAFT LEFT )
  111 FORMAT(80H DAY HOUR DAY HOUR ARRIVALS DEPARTURES
  1L ACTIVITY AT THE BASE )
  112 FORMAT(1H 14,15,18,15,113,116,114)
  113 FORMAT(27H1 AIRCRAFT AVAILABILITY - A6)
  114 FORMAT(14H0 DAY HOUR )
  115 FORMAT(1H 15,16)
C
C--> ADD LINE OF PRINTING IF DIMENSION EXCEEDED 6/12/75
116 FORMAT (/// 4H THE LAST LINE OF THE ABOVE TABLE MAY BE,
1 53HINACCURATE. ONE OR MORE INDICES EXCEEDED THE LENGTH,
2 13HOP THE TABLE. / 37HOPRINT LESS FREQUENTLY BY INCREASING,
3 44HTHE HOURS IN EACH INCREMENT IN CARD TYPE 00. )
  TOT = 0.0
  HTOT = 0
  DO 501 I = 1, 51

```

PCDL0010
PCDL0020
PCDL0030
PCDL0040
PCDL0050
PCDL0060
PCDL0070
PCDL0080
PCDL0090
PCDL0100
PCDL0110
PCDL0120
PCDL0130
PCDL0140
PCDL0150
PCDL0160
PCDL0170
PCDL0180
PCDL0190
PCDL0200
PCDL0210
PCDL0220
PCDL0230
PCDL0240
PCDL0250
PCDL0260
PCDL0270
PCDL0280
PCDL0290
PCDL0300
PCDL0310
PCDL0320
PCDL0330
PCDL0340
PCDL0350
PCDL0360
PCDL0370
PCDL0380
PCDL0390
PCDL0400
PCDL0410
PCDL0420
PCDL0430
PCDL0440
PCDL0450
PCDL0460
PCDL0470
PCDL0480
PCDL0490
PCDL0500
PCDL0510
PCDL0520
PCDL0530
PCDL0540
PCDL0550
PCDL0560


```

501 CALL ATRUZ (Q000HL,6H )
    POUT(I) = (+Q000HL)
    CALL ATRUZ (Q001HL,6H1 )
    POUT(1) = (+Q001HL)
    CALL ATRUZ (Q002HL,6H1 )
    POUT(51) = (+Q002HL)
    WRITE (6,100)
    WRITE (6,101)
    WRITE (6,106)
    DO 200 NST = 1, 200
    IF (WDEL(NST)) 180,180,250
180 IF (MDEL(NST)) 200,200,250
200 CONTINUE
    NST = 200
    NEND = NST
250 IF (NST - 1) 260,260,255
255 NST = NST - 1
260 DO 300 I = NST, 200
    IF (WDEL(I)) 270,270,280
270 IF (MDEL(I)) 300,300,280
280 NEND = I
300 CONTINUE
    DO 500 I = NST,NEND
    ZIN1 = I
    TIME = HRSINC * ZIN1
    NDAY = TIME / 24.0
    ZNDAY = NDAY
    NHR = TIME - ZNDAY * 24.0
    TOT = TOT + WDEL(I)
    MTOT = MTOT + MDEL(I)
    PCTC = (TOT/MTOT) * 100.0
    ZHT = MTOT
    ZNPPG = NPACT
    PCTH = (ZHT/ZNPPG)*100.0
    NPC = (PCTC + 3.0) / 2.0
    NPH = (PCTH + 3.0) / 2.0
    CALL ATRUZ (Q003HL,6H )
    POUT(NPH) = (+Q003HL)
    CALL ATRUZ (Q004HL,6H* )
    POUT(NPC) = (+Q004HL)
    WRITE (6,102) MTOT,PCTH,PCTC,NDAY,NHR,(POUT(K),K=1,51)
    CALL ATRUZ (Q005HL,6H )
    POUT(NPH) = (+Q005HL)
    CALL ATRUZ (Q006HL,6H )
    POUT(NPC) = (+Q006HL)
    CALL ATRUZ (Q007HL,6H1 )
    POUT(1) = (+Q007HL)
    CALL ATRUZ (Q008HL,6H1 )
    POUT(51) = (+Q008HL)
500 CONTINUE
    WRITE (6,106)
    IF (IVSP6) 800,800,700
    PRINT OFFLOAD BASE ACTIVITY
    C 700 WRITE (6,109)
    C 800 IFLAG WILL = 1 IF NDSP > 200
    C 8--> IFLAG=0

```

6/12/75

PCDL0570
PCDL0580
PCDL0590
PCDL0600
PCDL0610
PCDL0620
PCDL0630
PCDL0640
PCDL0650
PCDL0660
PCDL0670
PCDL0680
PCDL0690
PCDL0700
PCDL0710
PCDL0720
PCDL0730
PCDL0740
PCDL0750
PCDL0760
PCDL0770
PCDL0780
PCDL0790
PCDL0800
PCDL0810
PCDL0820
PCDL0830
PCDL0840
PCDL0850
PCDL0860
PCDL0870
PCDL0880
PCDL0890
PCDL0900
PCDL0910
PCDL0920
PCDL0930
PCDL0940
PCDL0950
PCDL0960
PCDL0970
PCDL0980
PCDL0990
PCDL1000
PCDL1010
PCDL1020
PCDL1030
PCDL1040
PCDL1050
PCDL1060
PCDL1070
PCDL1080
PCDL1090
PCDL1100
PCDL1110
PCDL1120
PCDL1130


```

CPDEL SUBROUTINE TO PRINT DELIVERIES
SUBROUTINE PDEL(NVT,TWT,NPA23,NEGC,POUT)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANCL(10),APAY(10,5),ARR(1000)
  1 AVID(10),CL(50),DIST(50,50),EPAX(10),FLOW(50),GRF(50),JHOW(10),
  2 LIST(5,100),NAME(50),NAMEP(50),NAMEV(200),NACTB(10,50),NBASE(50),
  3 NF(9),NPLG(50),NL(7),NO(50),NODE(50),NOV(200,11),NP(50),
  4 MPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBM(50,50),
  5 NTARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
  6 TB(50),THIN(50),VHT(200),VLENG(200),VND(200),VNT(200),WDEL(200),
  7 WTM(50),WTMV(2000),XL(50),XR(50)
  8 WDEL(200),WDELC(200),WDELC(200)
  DIMENSION POUT(51)
  DIMENSION MACARR(200),MACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBM
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANCL , APAY , AVID
  COMMON EPAX , GRF , JNOW , LISV , NAME , NAMEP
  COMMON NAMEV , MAOTB , NBASE , NF , NPLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1 , NU , NV , Q
  COMMON SPEED , TB , VHT , VLENG , VND , VNT
  COMMON WDEL , WTM , WTRV , GTEPH , WDEL
  COMMON WDELC , MDELC , HRSINC
  COMMON MACARR,MACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
  1 (XR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  100 FORMAT(59H1 DELIVERIES OF MATERIEL AND PERSONNEL FROM PRIORITY GROUP)
  101 PFORMAT(64H0 PERSONNEL PERCENT OF PERSONNEL PERCENT OF CARGO
  1AY HOUR
  102 FORMAT(1H I10, F20.2, F21.2, I7, I5, I1, 51A1)
  103 FORMAT(40H1 ITEMS IN PRIORITY GROUP NOT DELIVERED )
  106 FORMAT(1H 64K, 51H-----)
  1-----
  TOT = 0.0
  HTOT = 0
  DO 501 I = 1, 51
    CALL ATHEUZ (Q000HL,6H )
    POUT(I) = (+Q000HL)
    CALL ATHEUZ (Q001HL,6H1 )
    POUT(1) = (+Q001HL)
    CALL ATHEUZ (Q002HL,6H1 )
    POUT(51) = (+Q002HL)
    WRITE (6,100)HPJC
    WRITE (6,101)
    WRITE (6,106)
    DO 200 NST = 1, 200
      IF (WDEL(NST)) 180,180,250
      IF (WDEL(NST)) 200,200,250
    CONTINUE
    NST = 200
  250 NEND = NST

```



```

255 IF (NST - 1) 260,260,255
260 NST = NST - 1
260 DO 300 I = NST, 200
270 IF (WDEL(I)) 270,270,280
280 IF (MDEL(I)) 300,300,280
280 NEND = I
300 CONTINUE
DO 500 I = NST,NEND
ZIH1 = I
TIME = HRSINC * ZIH1
NDAY = TIME / 24.0
ZNDAY = NDAY
NHRS = TIME - ZNDAY * 24.0
TOT = TOT + WDEL(I)
MTOT = MTOT + MDEL(I)
PCTC = (TOT/TWT)*100.0
ZHI = HTOT
ZNPP3 = NPAP3
PCTH = (ZHT/ZNPP3)*100.0
NPC = (PCTC + 3.0) / 2.0
NPH = (PCTH + 3.0) / 2.0
CALL ATHRUZ (Q003HL,6H, )
POUT(NPH) = (+Q003HL)
CALL ATHRUZ (Q004HL,6H* )
POUT(NPC) = (+Q004HL)
WRITE (6,102)HCT,PCTH,PCTC,NDAY,NHRS,(POUT(K),K=1,51)
CALL ATHRUZ (Q005HL,6H )
POUT(NPH) = (+Q005HL)
CALL ATHRUZ (Q006HL,6H )
POUT(NPC) = (+Q006HL)
CALL ATHRUZ (Q007HL,6H1 )
POUT(1) = (+Q007HL)
CALL ATHRUZ (Q008HL,6H1 )
POUT(51) = (+Q008HL)
WDEL(I) = 0.0
MDEL(I) = 0
500 CONTINUE
WRITE (6,106)
WRITE (6,103)
IVSPI = 1
510 CALL PRP3 (NWT,TWT,NPAP3,IVSPI)
RETURN
END

```

PDEL0570
PDEL0580
PDEL0590
PDEL0600
PDEL0610
PDEL0620
PDEL0630
PDEL0640
PDEL0650
PDEL0660
PDEL0670
PDEL0680
PDEL0690
PDEL0700
PDEL0710
PDEL0720
PDEL0730
PDEL0740
PDEL0750
PDEL0760
PDEL0770
PDEL0780
PDEL0790
PDEL0800
PDEL0810
PDEL0820
PDEL0830
PDEL0840
PDEL0850
PDEL0860
PDEL0870
PDEL0880
PDEL0890
PDEL0900
PDEL0910
PDEL0920
PDEL0930
PDEL0940
PDEL0950
PDEL0960
PDEL0970
PDEL0980
PDEL0990

```

CPRPG      SUBROUTINE TO PRINT PRIORITY GROUP DATA
SUBROUTINE PRPG (NVT,TWT,NPAIPG,IVSP5)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANG(10),APAX(10,5),ARR(1000)
  1,AHID(10),CL(50),DIST(50,50),EPAX(10),FLOW(50),GRX(50),JNOH(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAOTB(10,50),NBASE(50),PRG0050
  3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NOH(50),MOV(200,11),NP(50),PRG0060
  4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBM(50,50),PRG0070
  5,NTRR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50),PRG0080
  6,TB(50),TMIN(50),VHT(200),VLENG(200),VVD(200),VWT(200),WDEL(200),PRG0090
  7,WTM(50),WTMT(2000),XL(50),XR(50),PRG0100
  8,WDEL(200),WDELCL(200),WDELCLC(200),PRG0110
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,KL,IR,DIST,NSUBM
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANG , APAX , ARR , AHID
  COMMON EPAX , GRX , JNOH , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTB , NBASE , NP , NPLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , THT , VLENG , VVD , VHT
  COMMON WDEL , WTH , WTHV , JTPPH , WDEL
  COMMON WDELCL , WDELCLC , HRSINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
  1 (KR(1),TMIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
101 FORMAT (28H0 VEHICLES IN PRIORITY GROUP )
102 FORMAT (69H0 NAME , NUMBER , WIDTH , HEIGHT , LENGTH , WEIGHT)
1F QUANTITY )
103 FORMAT (1H A6,I9,4F10.1,I11 )
104 FORMAT(20H0 CARGO WEIGHT IS , F10.2,6H TONS ,23H NUMBER OF PEP)
1RCONNEL IS I10)
105 FORMAT(17H0 CARGO COVERS , F10.1,48H THOUSAND SQUARE FEET OF FLOOR)
1R SPACE, AND WEIGHS , F10.2, 24H POUNDS PER SQUARE FOOT )
1F (IVSP5) 109,109,108
108 WRITE (6,101)
109 WRITE (6,102)
109 TWT = 0.0
109 TSP = 0.0
DO 200 I = 1, NVT
  IF (NSTK(16,I)) 200,200,150
  TWT = TWT + VHT(I) * FLOAT(NSTK(16,I))
  TSP = TSP + VVD(I)*VLENG(I)*FLOAT(NSTK(16,I))
  IF (IVSP5) 200,200,155
  155 WRITE (6,103)NAMEV(I),NV(I),VVD(I),VHT(I),VLENG(I), VWT(I), NSTK(1)
  16,I)
200 CONTINUE
  NPAIPG = NSTK(16,1)
  ZNPX = NPAIPG
  TWT = TWT - ZNPX * VWT(1)
  TSP = TSP - (ZNPX*VVD(1)*VLENG(1))
  TWT = TWT / 2000.
  TSP = TSP /144.

```

PRPG0570
PRPG0580
PRPG0590
PRPG0600
PRPG0610
PRPG0620

DENS = TWT / TSPP
TSPP = TSPP/1000.
WHITE (6,104) TWT, NPA1PG
WHITE (6,105) TSPP, DENS
RETURN
END


```

CRDNT      SUBROUTINE TO READ NETWORK DATA
SUBROUTINE RDNT (NModes)
  DIMENSION AHT(10),ALENG(10),ANAME(10),AWO(10),APAY(10,5),ARE(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAL(10),FLOV(50),GET(50),JNOW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAOTS(200),NAOTE(10,50),NBASE(50),RDNT0050
  3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),RDNT0060
  4,NPAC(10,50),NRUT(50),NSAVE(50),NSTA(16,200),NSUBN(50,50),RDNT0070
  5,NTABR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50),RDNT0080
  6,TB(50),TMIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),RDNT0090
  7,WTH(50),WTHV(2000),XL(50),XR(50)RDNT0100
  8,WDEL(200),WDEL1(200),WDEL2(200)RDNT0110
  DIMENSION ZLONG(50),ZLAT(50)RDNT0120
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)RDNT0130
  DIMENSION FILL(250)RDNT0140
  REAL*8 ANAME,NAME,NAMEP
  REAL*8 NAMEP
  COMMON STA,WTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , AWO , APAY , AWID
  COMMON EPAX , GET , JNOW , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTS , NBASE , NP , NPLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VWD , VWT
  COMMON WDEL , WTH , WTHV , WTPPH , WDEL
  COMMON WDEL1 , WDEL2 , HRSINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),ZLOW(1)),(WTL(1),NRUT(1)),(XL(1),CL(1)),
  1 (XR(1),TMIN(1)), (DIST(1),ARE(1)), (DIST(1001),NTABR(1)),
  2 (DIST(2002),NSTK(1))
  1/13/76
  C-->      CHANGE FORMAT TO CORRESPOND TO CARD TYPE 03 DESCRIPTION
  C-->
  C-101  FORMAT (I2,X1,A6,I5,F5.0,F7.0,F3.0,F7.0,F3.0)
  101  FORMAT (I2,X1,A6,I5,F5.0,F7.0,I1,F2.0,F7.0,I1,F2.0)
  102  FORMAT (I2, 3I, 2I5, 2F5.0 )
  103  FORMAT (38HOPROGRAM STOPPED MORE THAN 50 NODES )
  104  FORMAT (1H 30I,A6,I8,F13.2,F10.0,F9.0,F13.0,F10.0 )
  105  FORMAT (27H1 BASES USED IN DEPLOYMENT )
  106  FORMAT(1H030I,71H NAME NUMBER GROUND TIME DEG LAT MIN LAT
  1  DEG LONG MIN LONG )
  107  FORMAT (1H )
  108  FORMAT (13H1 LINKS INPUT )
  109  FORMAT(30I,32H FROM TO DISTANCE )
  110  FORMAT(1H 30I,A6,5I,A6,F14.1)
  111  FORMAT (1H 30I,2I5,20I,42HLINK REJECTED - BASE MISSING IN BASE LISRDNT0460
  112  FORMAT (1H 30I,2I5,20I,42HLINK REJECTED - BASE MISSING IN BASE LISRDNT0460
  113  FORMAT (1H 30I,A6,5I,A6,17H LINK DENIED )
  114  FORMAT (1H 30I,A6,5I,A6,17H LINK DENIED )
  C      READ NODE DATA
  I = 1
  WRITE (6,105)
  WRITE (6,106)
  WRITE (6,107)
  150  READ (5,101)ISE,NAME(I),NO(I),GET(I),ZLATD,ZLATM,ZLONGD,ZLONGH
  IF (ISE - 3) 164,160,164
  160  WRITE (6,104)NAME(I),NO(I),GET(I),ZLATD,ZLATM,ZLONGD,ZLONGH
  C
  1/13/76
  C-->
  C-->      IF DEGREE FIELD IS NEGATIVE, SET MINUTES NEGATIVE
  C-->

```


RDNT1220
RDNT1230
RDNT1240
RDNT1250
RDNT1260
RDNT1270
RDNT1280
RDNT1290
RDNT1300
RDNT1310
RDNT1320
RDNT1330
RDNT1340
RDNT1350
RDNT1360
RDNT1370
RDNT1380
RDNT1390
RDNT1400
RDNT1410
RDNT1420
RDNT1430
RDNT1440
RDNT1450
RDNT1460
RDNT1470
RDNT1480
RDNT1490

```
365 CONTINUE
370 CONTINUE
C   READ LINK DATA
210 READ (5,102) ISW, NFR, NTO, DISTX
    IF (ISW - 4) 500, 215, 500
215 IF (NFR - NTO) 220, 216, 216
216 NIZ = NTO
    NTO = NFR
    NFR = NIZ
C   CONVERT TO NODE NUMBERS
220 DO 240 I = 1, NNODES
    IF (NO(I) - NFR) 240, 250, 240
240 CONTINUE
    GO TO 301
250 DO 260 J = 1, NNODES
    IF (NO(J) - NTO) 260, 270, 260
260 CONTINUE
301 WRITE (6,112) NFR, NTO
    GO TO 210
270 IF (DISTX) 400, 400, 410
400 DIST(I,J) = 999999.0
    WRITE (6,114) NAME(I), NAME(J)
    GO TO 210
410 DIST(I,J) = DISTX
    WRITE (6,111) NAME(I), NAME(J), DISTX
    GO TO 210
500 RETURN
END
```



```

CDDP3      SUBROUTINE TO READ DATA ON A PRIORITY GROUP
          SUBROUTINE RDPG (MVS,ISW,MFGC)
          DIMENSION AHT(10),ALEN(10),ANAME(10),ANO(10),APAY(10,5),ARR(1000)
          1,AWID(10),CL(50),DIST(50,50),EPAI(10),FLOW(50),GET(50),JHOW(10),
          2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAOTB(10,50),NBASE(50),
          3,NP(9),NPLG(50),NL(7),NO(50),MODE(50),NON(50),NOV(200,11),NP(50),
          4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
          5,NTABR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),SPA(50)
          6,TB(50),TMIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
          7,WTH(50),WTHV(2000),XL(50),XR(50)
          8,WDEL(200),WDEL1(200),WDEL2(200),NORMAN(10)
          DIMENSION MACABR(200),MACDEP(200),MACDEL(200)
          DIMENSION FILL(250)
          REAL*8 ANAME,NAME,NAMEV
          REAL*8 NAMEP
          COMMON STA,NTL,IL,XR,DIST,NSUBN
          COMMON FILL
          COMMON AHT , ALEN3 , ANAME , ANO , APAY , AWID
          COMMON EPAI , GET , JHOW , LISV , NAME , NAMEP
          COMMON NAMEV , NAOTB , NBASE , NP , NPLG , NL
          COMMON NO , NODE , NON , NOV , NP , NPAC
          COMMON NSAVE,NTL1,NU,NV,Q
          COMMON SPEED , TB , VHT , VLENG , VWD , VWT
          COMMON WDEL , WTH , WTHV , GTPFH , WDEL
          COMMON WDEL1 , WDEL2 , HRSINC
          COMMON MACABR,MACDEP,NORMAN
          EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
          1 (IR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTABR(1)),
          2 (DIST(2002),NSTK(1))
          99 FORMAT (43HOPROGRAM STOPPED - DATA CARDS OUT OF ORDER )
          100 FORMAT (17H1 PRIORITY GROUP I4)
          101 FORMAT (I2,3X,9I2,P7,D,2I5)
          103 FORMAT (6H FLAGS 24X,25H QUANTITY UNIT LIST )
          104 FORMAT (1H 2X,9I3,P10,2,I6,I8)
          NNEW = 0
          DO 140 IZ = 1,NVT
          140 NSFR(16,IZ) = 0
          150 READ(5,101) ISW, (WFSP(I),I=1,9),Q(1),NU(1),NL(1)
          170 IF (ISW - 8) 170,200,170
          200 IZ = 16
          201 IF (ISW) 201,201,175
          201 WRITE(6,100) MFGC
          NNEW = 1
          DO 180 I = 1,9
          180 NF(I) = WFSP(I)
          WRITE (6,104) (NF(I),I=1,9), Q(1), NU(1),NL(1)
          CALL SFLS (IZ,NVT,NNEW)
          GO TO 150
          500 RETURN
          END

```

RDPG0010
 RDPG0020
 RDPG0030
 RDPG0040
 RDPG0050
 RDPG0060
 RDPG0070
 RDPG0080
 RDPG0090
 RDPG0100
 RDPG0110
 RDPG0120
 RDPG0130
 RDPG0140
 RDPG0150
 RDPG0160
 RDPG0170
 RDPG0180
 RDPG0190
 RDPG0200
 RDPG0210
 RDPG0220
 RDPG0230
 RDPG0240
 RDPG0250
 RDPG0260
 RDPG0270
 RDPG0280
 RDPG0290
 RDPG0300
 RDPG0310
 RDPG0320
 RDPG0330
 RDPG0340
 RDPG0350
 RDPG0360
 RDPG0370
 RDPG0380
 RDPG0390
 RDPG0400
 RDPG0410
 RDPG0420
 RDPG0430
 RDPG0440
 RDPG0450
 RDPG0460
 RDPG0470
 RDPG0480
 RDPG0490
 RDPG0500
 RDPG0510
 RDPG0520
 RDPG0530
 RDPG0540

```

CRDST      SUBROUTINE TO READ STOCK DATA
          SUBROUTINE RDST(NSTL,NVT,NNODES)
          DIMENSION AHT(10),ALENG(10),ANAME(10),ANQ(10),APAY(10),ARR(1000)
          1,AWID(10),CL(50),DIST(50,50),EPAL(10),FLCM(50),GR(50),JNOW(10),
          2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NAOTB(10,50),NBASE(50),
          3,NF(9),NFLG(50),NL(7),NO(50),NODE(50),NOV(50),NOY(200,11),NP(50),
          4,NPAC(10,50),NR(50),NR(50),NSAVE(50),NSTK(16,200),NSUBM(50,50),
          5,NPARR(1000),NTL(50),NTL1(50),NU(7),NV(200),O(7),SPEED(10),STA(50)
          6,TB(50),TMIN(50),VHT(200),VLEN3(200),VND(200),VNT(200),WDEL(200),
          7,WTH(50),WTHV(2000),XL(50),XR(50)
          8,WDEL(200),WDELC(200),WDELC(200)
          DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
          DIMENSION FILL(250)
          REAL*8 ANAME,NAME,NAMEV
          REAL*8 NAMEP
          COMMON STA,NTL,XL,XR,DIST,NSUBN
          COMMON FILL
          COMMON AHT, ALENG, ANAME, ANQ, APAY, AWID
          COMMON EPAX, GR, JNOW, LISV, NAME, NAMEP
          COMMON NAMEV, NAOTB, NBASE, NP, NTLG, NL
          COMMON NO, NODE, NOV, NPAC
          COMMON NSAVE, NTL1,NU,NV,Q
          COMMON SPEED, TB, VHT, VLEN3, VND, VNT
          COMMON WDEL, WTH, WTHV, GTPPH, WDEL
          COMMON WDELC, MDELC, HRSINC
          COMMON NACARR,NACDEP,NORMAN
          EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NR(1)),(XL(1),CL(1)),
          1 (IR(1),THIN(1)),(DIST(1),ARR(1)),(DIST(1001),WARR(1)),
          2 (DIST(2002),NSTK(1))
          99 FORMAT(44H)PROGRAM STOPPED - MORE THAN 15 STOCK LISTS
          101 FORMAT(12,3I,2I5,F5.0,2I5)
          102 FORMAT(40HC FLAG BASE QUANTITY UNIT LIST )
          103 FORMAT(19H1 STOCK LISTS )
          104 FORMAT(1H 14,2X,A6,F11.2,I6,I8)
          105 FORMAT(29H LIST REJECTED - BASE I5,18H NOT IN BASE LIST )
          WRITE(6,103)
          WRITE(6,102)
          DO 140 I = 1,16
          DO 140 J = 1, NVT
          140 NSTK(I,J) = 0
          NPLS = 0
          NBL = 0
          IZ = 0
          150 READ(5,101) ISH,NFX,NBX,O(1),NU(1),NL(1)
          IF (ISH - 7) 500,200,500
          200 IF (NFX - NPLS) 205,201,205
          201 IF (NBX - NBL) 205,210,205
          205 IZ = IZ + 1
          NNEW = 1
          210 IF (IZ - 15) 220,220,999
          220 DO 250 M = 1, NNODES
          250 IF (NO(M) - NBX ) 250,260,250
          CONTINUE
          WRITE(6,105)NBX
          IF (NNEW) 150,150,256
          IZ = IZ - 1
          RDST0010
          RDST0020
          RDST0030
          RDST0040
          RDST0050
          RDST0060
          RDST0070
          RDST0080
          RDST0090
          RDST0100
          RDST0110
          RDST0120
          RDST0130
          RDST0140
          RDST0150
          RDST0160
          RDST0170
          RDST0180
          RDST0190
          RDST0200
          RDST0210
          RDST0220
          RDST0230
          RDST0240
          RDST0250
          RDST0260
          RDST0270
          RDST0280
          RDST0290
          RDST0300
          RDST0310
          RDST0320
          RDST0330
          RDST0340
          RDST0350
          RDST0360
          RDST0370
          RDST0380
          RDST0390
          RDST0400
          RDST0410
          RDST0420
          RDST0430
          RDST0440
          RDST0450
          RDST0460
          RDST0470
          RDST0480
          RDST0490
          RDST0500
          RDST0510
          RDST0520
          RDST0530
          RDST0540
          RDST0550
          RDST0560

```

RDST0570
RDST0580
RDST0590
RDST0600
RDST0610
RDST0620
RDST0630
RDST0640
RDST0650
RDST0660
RDST0670
RDST0680
RDST0690
RDST0700
RDST0710

260 GO TO 150
 NPLS = NPI
 WRITE (6,104) NPL, NAME(H),Q(1),NU(1),NL(1)
 NBASE (IZ) = M
 NPLG (IZ) = NPI
 CALL SFLS (IZ,NVT,NNEW)
 NNEW = 0
500 GO TO 150
 NSPL = IZ
999 RETURN
 WRITE (6,99)
 CALL EXIT
 RETURN
 END


```

CSB2 SUBROUTINE TO FIND ROUTES OUT AND BACK
SUBROUTINE SBN2 (NOL,IWAR)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANJ(10),APAY(10,5),ARR(1000)
  1,AWID(10),CL(50),DIST(50,50),EPAY(10),FLOW(50),GR(50),JNOW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAOTB(10,50),NBASE(50),
  3,NR(9),NPLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NRUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NRAB(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(17),SPEED(10),STA(50)
  6,TB(50),TMIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
  7,WPH(50),WTHV(2000),XL(50),XR(50)
  8,WDEL(200),WDELCL(200),HDELCL(200)
  DIMENSION MACARR(200),MACDEP(200),NORHAN(10)
  DIMENSION FILL(250)
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AWID
  COMMON EPAX , JRT , JNOW , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTB , NBASE , NP , NPLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VWD , VWT
  COMMON WDEL , WTH , WTHV , GTPFH , NDEL
  COMMON WDELCL , MDELCL , HESINC
  COMMON MACARR,MACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NRUT(1)),(XL(1),CL(1)),
  1 (XR(1),TMIN(1)),(DIST(1),ARR(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  100 FORMAT (19HMINIMUM TIME FROM A6,4H TO A6,4H IS F10.2 )
  101 FORMAT (10H ROUTE IS )
  102 FORMAT (1H 10(A6.6X) )
  103 FORMAT (25HMAXIMUM-FLOW ROUTE FROM A6,4H TO A6,4H IS )
  C FIND ROUTE BACK
  I = NOL
  DO 200 J = 1, 50
    NRUT(J) = I
    IF (I - IWAR) 180,210,180
  180 I = NSAVE(I)
  200 CONTINUE
  GO TO 250
  210 NRUT(J+1) = -1
  C PRINT ROUTE BACK
  250 IX = 1
  DO 300 I = 1, J
    NI = J + 1 - I
    ND = NRUT(NI)
    NAMEP(IX) = NAME(ND)
    IX = IX + 1
  300 CONTINUE
  WRITE (6,100)NAME(IWAR), NAME(NOL), TB(NOL)
  WRITE (6,101)
  WRITE (6,102) (NAMEP(I),I = 1,J )
  C FIND ROUTE OUT
  I = NOL
  DO 400 J = 1, 50

```

SBN20010
 SBN20020
 SBN20030
 SBN20040
 SBN20050
 SBN20060
 SBN20070
 SBN20080
 SBN20090
 SBN20100
 SBN20110
 SBN20120
 SBN20130
 SBN20140
 SBN20150
 SBN20160
 SBN20170
 SBN20180
 SBN20190
 SBN20200
 SBN20210
 SBN20220
 SBN20230
 SBN20240
 SBN20250
 SBN20260
 SBN20270
 SBN20280
 SBN20290
 SBN20300
 SBN20310
 SBN20320
 SBN20330
 SBN20340
 SBN20350
 SBN20360
 SBN20370
 SBN20380
 SBN20390
 SBN20400
 SBN20410
 SBN20420
 SBN20430
 SBN20440
 SBN20450
 SBN20460
 SBN20470
 SBN20480
 SBN20490
 SBN20500
 SBN20510
 SBN20520
 SBN20530
 SBN20540
 SBN20550
 SBN20560

SBN20570
SBN20580
SBN20590
SBN20600
SBN20610
SBN20620
SBN20630
SBN20640
SBN20650
SBN20660
SBN20670
SBN20680
SBN20690
SBN20700
SBN20710
SBN20720
SBN20730

```

      NODE(J) = I
      IF (I - IWAR) 380,410,380
380   I = NSUBN(NOL, I)
400   CONTINUE
      GO TO 450
410   NODE(J+1) = - 1
      C PRINT ROUTE OUT
450   IX = 1
      DO 500 I = 1, J
      ND = NODE(I)
      NAMEP(IX) = NAME(ND)
      IX = IX + 1
500   CONTINUE
      WRITE (6,103) NAME(NOL), NAME(IWAR)
      WRITE (6,102) (NAMEP(I), I=1,J )
750   RETURN
      END
```

```

CSILS      SUBROUTINE TO STORE LIST OF VEHICLES
SUBROUTINE STLS (IZ,NVT,MNEW)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10.5),AER(1000)
  1,AWID(10),CL(50),DIST(50.50),EPAX(10),FLOW(50),GR(50),JNOW(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAOTB(200),NAOTB(10,50),NBASE(50),
  3,NF(9),NPLG(50),NL(7),NO(50),NODE(50),NOV(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NR(50),NR(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NPAC(1000),NTL(50),NTL(50),NU(7),NV(200),Q(7),SPEED(10),SPA(50)
  6,IB(50),THIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
  7,WTM(50),WTMV(2000),XL(50),XR(50)
  8,WDEL(200),WDEL(200),NACDEP(200),NORMAN(10)
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEP
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AWID
  COMMON EPAX , GR , JNOW , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTB , NBASE , NF , NPLG , NL
  COMMON NO , NODE , NOV , NP , NPAC
  COMMON NSAVE,NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VWD , VWT
  COMMON WDEL , WTM , WTMV , GTPPH , MDEL
  COMMON WDEL , MDEL , HESINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NR(1)),(XL(1),CL(1)),
  1 (KR(1),TMIN(1)),(DIST(1),AER(1)),(DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  106 FORMAT(28H PORTION USING VEHICLE LIST 15,34H REJECTED - LIST NUMBERSTLS0300
  1R EXCEEDS 5 )
  107 FORMAT(20H PORTION USING UNIT 15,44H REJECTED - UNIT NUMBER MUST BSL0320
  1Z FROM 1 TO 11 )
  301 NUX = NU(1)
  IF (NUX) 388,388,302
  302 IF (NUX - 11) 303,303,388
  388 WRITE (6,107)NUX
  460 IF (MNEW) 450,450,465
  465 IZ = IZ - 1
  GO TO 400
  303 QTY = Q(1)
  IF (NL(1)) 309,320,329
  309 NLX = -NL(1)
  IF (NLX - 5) 310,310,399
  399 WRITE (6,106)NLX
  IF (MNEW) 450,450,465
  GO TO 400
  500 DO 315 K = 1,NVT
  310 DO 312 L = 1,100
  IF (LISV(NLX,L)) 313,313,311
  311 IF (LISV(NLX,L) - K) 312,315,312
  312 CONTINUE
  313 NA = QTY * FLOAT(NOV(K,NUX)) + .99
  NSTK(IZ,K) = NSTK(IZ,K) + NA
  315 CONTINUE
  GO TO 400

```

STLS0010
STLS0020
STLS0030
STLS0040
STLS0050
STLS0060
STLS0070
STLS0080
STLS0090
STLS0100
STLS0110
STLS0120
STLS0130
STLS0140
STLS0150
STLS0160
STLS0170
STLS0180
STLS0190
STLS0200
STLS0210
STLS0220
STLS0230
STLS0240
STLS0250
STLS0260
STLS0270
STLS0280
STLS0290
STLS0300
STLS0310
STLS0320
STLS0330
STLS0340
STLS0350
STLS0360
STLS0370
STLS0380
STLS0390
STLS0400
STLS0410
STLS0420
STLS0430
STLS0440
STLS0450
STLS0460
STLS0470
STLS0480
STLS0490
STLS0500
STLS0510
STLS0520
STLS0530
STLS0540
STLS0550
STLS0560

STLS0570
STLS0580
STLS0590
STLS0600
STLS0610
STLS0620
STLS0630
STLS0640
STLS0650
STLS0660
STLS0670
STLS0680
STLS0690
STLS0700
STLS0710
STLS0720
STLS0730
STLS0740

```

320 DO 325 K = 1, NVT
325 NA = QTY * FLOAT(NCV(K, NUX)) + .99
    NSTK(IZ, K) = NSTK(IZ, K) + NA
    GO TO 400
329 NLX = NL(1)
    IF (NLX - 5) 330, 330, 399
330 DO 335 K = 1, NVT
    DO 332 L = 1, 100
331 IF (LISTV(NLX, L)) 335, 335, 331
332 IF (LISTV(NLX, L) - K) 332, 334, 332
332 CONTINUE
    GO TO 335
334 NA = QTY * FLOAT(NCV(K, NUX)) + .99
    NSTK(IZ, K) = NSTK(IZ, K) + NA
335 CONTINUE
400 CONTINUE
450 RETURN
    END

```



```

CTMBK      SUBROUTINE TO COMPUTE TIME BACK FROM WAR
SUBROUTINE THBK (IWAR,SPEEDX,RANGE,RNODES,NBDN)
  DIMENSION AHT(10),ALENG(10),ANAME(10),AND(10),APAY(10,5),ARR(1000)
  1,AVID(10),CL(50),DIST(50,50),EPAX(10),FLOW(50),GET(50),JMOV(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEV(200),NACTB(10,50),NBASE(50),
  3,NF(9),NFLG(50),NL(7),NO(50),NODE(50),NON(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NFARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
  6,TB(50),THIN(50),VHT(200),VLENG(200),VWD(200),VWT(200),WDEL(200),
  7,WFM(50),WTMV(200),XL(50),XR(50)
  8,WDEL(200),WDELCL(200),WDELCLC(200)
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,AL,IR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ARO , APAY , AVID
  COMMON EPAX , GET , JNCW , LISV , NAME , NAMEP
  COMMON NAMEV , NACTB , NBASE , NF , NFLG , NL
  COMMON NO , NODE , NON , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q , VHT , VLENG , VWD , VWT
  COMMON SPEED , TB , WFM , WTMV , WDEL , WDELCL
  COMMON WDELCLC , WDELCLC , HESINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)), (NTL(1),NROUT(1)), (XL(1),CL(1)),
  1 (IR(1),THIN(1)), (DIST(1),ARR(1)), (DIST(1001),NTARR(1)),
  2 (DIST(2002),NSTK(1))
  C
  C-->      CHANGE FORMAT STATEMENT      6/12/75
C101      FORMAT (5I, 15,15)
101      FORMAT (5X, 15I5)
102      FORMAT (14H DENIED BASES 2C15)
      NP1 = NNODES + 1
      NODE(1) = IWAR
      IF (NBDN) 106,106,105
105      READ (5,101)(NSAVE(I),I=1,NBDN)
106      WRITE (6,102)(NSAVE(I),I=1,NBDN)
      IZ = 2
      DO 200 I = 2, NP1
      IX = NNODES + 2 - I
      IF (IX - IWAR) 140,200,140
140      IF (NBDN) 150,150,141
141      DO 145 K = 1,NBDN
      IF (NO(IX) - NSAVE(K)) 145,200,145
145      CONTINUE
150      NODE(IZ) = IX
      IZ = IZ + 1
200      CONTINUE
      WINDF = - 1.0
      IF (IZ - 50) 250,250,300
250      NODE(IZ) = -1
300      CALL HINT(RANGE,SPEEDX,WINDF,THINK,TGDEST,CLX)
      RETURN
      END

```

THBK0010
 THBK0020
 THBK0030
 THBK0040
 THBK0050
 THBK0060
 THBK0070
 THBK0080
 THBK0090
 THBK0100
 THBK0110
 THBK0120
 THBK0130
 THBK0140
 THBK0150
 THBK0160
 THBK0170
 THBK0180
 THBK0190
 THBK0200
 THBK0210
 THBK0220
 THBK0230
 THBK0240
 THBK0250
 THBK0260
 THBK0270
 THBK0280
 THBK0290
 THBK0300
 THBK0310
 THBK0320
 THBK0330
 THBK0340
 THBK0350
 THBK0360
 THBK0370
 THBK0380
 THBK0390
 THBK0400
 THBK0410
 THBK0420
 THBK0430
 THBK0440
 THBK0450
 THBK0460
 THBK0470
 THBK0480
 THBK0490
 THBK0500
 THBK0510
 THBK0520
 THBK0530
 THBK0540
 THBK0550
 THBK0560


```

CWPIM      SUBROUTINE TO FIND WEIGHT-TIME TABLE
SUBROUTINE WTHM(IAC)
  DIMENSION AHT(10),ALENG(10),ANAME(10),ANO(10),APAY(10,5),AER(1000)
  1,AUID(10),CL(50),DIST(50,50),EPAX(10),FLCH(50),GRX(50),JMOV(10),
  2,LISV(5,100),NAME(50),NAMEP(50),NAMEZ(200),NAOTB(10,50),NBASE(50),
  3,NP(9),NPLG(50),NL(7),NO(50),NODE(50),NOV(200,11),NP(50),
  4,NPAC(10,50),NROUT(50),NSAVE(50),NSTK(16,200),NSUBN(50,50),
  5,NFARR(1000),NTL(50),NTL1(50),NU(7),NV(200),Q(7),SPEED(10),STA(50)
  6,TB(50),TMIN(50),VHT(200),VLENG(200),VND(200),VHT(200),WDEL(200),
  7,WFM(50),WTMT(2000),XL(50),XR(50)
  8,WDEL(200),WDELC(200),WDELC(200)
  DIMENSION NACARR(200),NACDEP(200),NORMAN(10)
  DIMENSION FILL(250)
  REAL*8 ANAME,NAME,NAMEV
  REAL*8 NAMEP
  COMMON STA,NTL,XL,XR,DIST,NSUBN
  COMMON FILL
  COMMON AHT , ALENG , ANAME , ANO , APAY , AUID
  COMMON EPAX , GRX , JMOV , LISV , NAME , NAMEP
  COMMON NAMEV , NAOTB , NBASE , NF , NPLG , NL
  COMMON NO , NODE , NOV , NP , NPAC
  COMMON NSAVE , NTL1,NU,NV,Q
  COMMON SPEED , TB , VHT , VLENG , VND , VHT
  COMMON WDEL , WTM , WTMV , GTPPH , WDEL
  COMMON WDELC , WDELC , HRSINC
  COMMON NACARR,NACDEP,NORMAN
  EQUIVALENCE (STA(1),FLOW(1)),(NTL(1),NROUT(1)),(XL(1),CL(1)),
  1 (XR(1),TMIN(1)),(DIST(1),AER(1)),(DIST(1001),NFARR(1)),
  2 (DIST(2002),NSTK(1))
  WINDP = 1.0
  I = 1
  CLN = APAY(IAC,5)
  150 CALL MINT(CLN, SPEED(IAC), WINDP,TMINX,TGDEST,CLX)
  IF (CLX) 155,155,160
  155 WTH(I) = - 1
  RETURN
  160 WTH(I) = PAYL(IAC,CLX)
  WTH(I+1) = TMINX
  IF (CLX - APAY(IAC,1) ) 300,300,200
  200 IF (I - 49) 250,301,301
  250 I = I + 2
  CLN = CLX - 1.0
  GO TO 150
  300 WTH(I + 2) = - 1.0
  301 RETURN
  END

```

WTHM0010
 WTHM0020
 WTHM0030
 WTHM0040
 WTHM0050
 WTHM0060
 WTHM0070
 WTHM0080
 WTHM0090
 WTHM0100
 WTHM0110
 WTHM0120
 WTHM0130
 WTHM0140
 WTHM0150
 WTHM0160
 WTHM0170
 WTHM0180
 WTHM0190
 WTHM0200
 WTHM0210
 WTHM0220
 WTHM0230
 WTHM0240
 WTHM0250
 WTHM0260
 WTHM0270
 WTHM0280
 WTHM0290
 WTHM0300
 WTHM0310
 WTHM0320
 WTHM0330
 WTHM0340
 WTHM0350
 WTHM0360
 WTHM0370
 WTHM0380
 WTHM0390
 WTHM0400
 WTHM0410
 WTHM0420
 WTHM0430
 WTHM0440
 WTHM0450
 WTHM0460

```
ATHRUZ  CSECT  (14,12)
        SAVE  OH
        STM  14,12,12(13)  SAVE REGISTERS
        BALR 2,0
        USING *,2
        ST   13,SAVE*4
        LA   13,SAVE
        L    3,0(0,1)
        L    4,4(0,1)
        MVC  0(4,3),0(4)
        L    13,SAVE*4
        RETURN (14,12)
        LH   14,12,12(13)  RESTORE THE REGISTERS
        BR   14,RETURN
        DS   18F
        END
```

BIBLIOGRAPHY

Ford, L. R., and D. R. Fulkerson, *Flows in Networks*, The Rand Corporation, R-375-PR, August 1962.

Hayes, J. H., *Future Army Deployment Requirements* (U), The Rand Corporation, R-1673-PR, April 1975 (Confidential).

Sharpe, W. F., *The Army Deployment Simulator*, The Rand Corporation, RM-4219-ISA, March 1965.